January 1972

Limitations of Some Finite Difference Methods in Solving the Strongly Nonlinear Equation of Unsaturated Flow in Soils

Roland W. Jeppson

LIMITATIONS OF SOME FINITE DIFFERENCE METHODS IN SOLVING THE STRONGLY NONLINEAR EQUATION OF UNSATURATED FLOW IN SOILS

by

Roland W. Jeppson

The work reported herein was carried out under cooperative agreement no. 12-14-100-9715(41) between the Utah Water Research Laboratory and the Agricultural Research Service with funds from the Soil and Water Conservation Research Division, Northwest Watershed Research Center, Boise, Idaho 83701.

Utah Water Research Laboratory
College of Engineering
Utah State University
Logan, Utah 84322

September 1972

PRWG-59c-8

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATION

a_1, a_2, a_3, a_4, a_5 - coefficients whose magnitude depends on ξ

b_1, b_2, c - coefficients not dependent on ξ

D - dimensionless depth

$\mathbf{D}$ - Jacobian

F - finite difference operator

g - acceleration of gravity

G - finite difference operator

h - dimensionless hydraulic head

h_o - initial dimensionless hydraulic head

H - finite difference operator

i, j - subscripts denoting grid point

k - superscript denoting time step

K - hydraulic conductivity

K_o - saturated hydraulic conductivity

K_r - relative hydraulic conductivity

M_r - subscript denoting grid point just in head of wetting front in r-direction

M_z - subscript denoting grid point just in head of wetting front in negative z-direction

N_r - subscript denoting number of grid lines from axis of symmetry to outside radius of problem

N_z - subscript denoting number of grid lines from surface to impervious layer

p - pressure

p_b - bubbling pressure or characteristic length when divided by ρg used to nondimensionalize length variables of the problem

p_h - capillary pressure head $p_h = p/\rho g$

p_t - dimensionless pressure

r - dimensionless radial coordinate

S - saturation

S_e - effective saturation

S_r - residual saturation

$|v|/K_o$ - dimensionless rate of application

z - dimensionless axial coordinate

η - porosity

λ - pore size distribution exponent

ξ - transformed dependent variable

ρ - fluid density

τ - dimensionless time parameter

INTRODUCTION

During the past decade as the high speed digital computer has become generally accessible to all researchers, an increasing number of numerical solutions have appeared in the literature to complex the difficult problems formulated in terms of partial differential equations. Since nature, with few exceptions is nonlinear, this activity has rapidly progressed from solving problems associated with linear partial differential equations, to those associated with nonlinear equations. This progress is occurring without an extant theory for these nonlinear equations. Prominent among the numerical methods being used, are the methods of finite differences. The finite difference schemes used for the nonlinear equations consist principally of extensions of those methods developed for, and whose performance has been mathematically analyzed for, solving problems associated with linear partial differential equations. While the extensions do solve the resulting system of algebraic equations little is known concerning the convergence of stability of the method. Just because a method is unconditionally stable in solving a linear equation, is no guarantee that the same will be true for a nonlinear problem. Indeed, each of the multitude of nonlinearities that can exist may cause its own unique numerical difficulties. The validity of the solution method can only be implied by noting whether the solution is in agreement with what is known from physical observations of the problem.

The field of numerical solution to water movement in unsaturated soils has experienced the rapid development mentioned above. The recent book by Remson, Hornberger and Molz (1971) cites a large number of references dealing with the numerical schemes for solving such problems. As Brutsaert (1971) points out, however, all two-dimensional solutions, with only some unpublished exceptions at the time of his paper, are restricted to situations in which saturations vary smoothly in both space and time, and consequently do not contain sharp wetting fronts. A number of authors have eluded to numerical difficulties, particularly if two spare coordinates were involved. Undoubtedly, many workers have encountered difficulties, that in some cases even prevented solutions, that are not mentioned in published or unpublished literature. In studying the transient saturated-unsaturated flow in a rectangular region, Verna and Brutsaert (1970) found that the performance of common implicit schemes was at best marginal and unacceptable as more of the regions become unsaturated. While they conclude that the major source of difficulty is associated with locating the position of the boundary separating the saturated and unsaturated zones, they also found nonconvergence which no doubt is a consequence of the nonlinearities of the partial differential equations. In Freeze's (1971) noteworthy model of an entire saturated-unsaturated groundwater basin, numerical difficulties are noted due to oscillations in the predicted and calculated values of soil moisture tension which cause the coefficient in the finite difference equations which depend upon the pressure to oscillate. In solving the one-dimensional infiltration problem, Smith and Woolhiser (1971) noted similar oscillations of the pressure within an iteration in determining the magnitude of coefficients. After studying the general nature of these oscillations, they terminate their iteration by using a weighted average of pressures from the final two iterations. These described difficulties are directly related to the strong nonlinearities in the equation of flow. It is believed, in addition to these difficulties associated with the convergence of an iterative solution method, that the nonlinearities can cause scatter in such solution results as infiltration curves. Furthermore, as will be demonstrated later, considerably different finite difference solutions will occur from minor but invalid changes in the type of approximations used. Not only does this state of affairs emphasize the need for meticulous concern in appropriately differencing the flow equation, the selection of the finite difference scheme used, and means of solving the resulting system of algebraic equation, but it points out that more complete theories of methods of solving initial-boundary-value problems associated with strongly nonlinear equations such as the equation of flow, are needed.

The remarks in this paper will be restricted to the hydraulic head based equation of flow, but one might expect similar behavior from the diffusivity form of the equation of flow. To illustrate some of these items, both the transient problem of one-dimensional vertical moisture movements, and the transient problem of three-dimensional but axisymmetric (and therefore, actually two-dimensional) moisture movement from infiltration applied on a circular surface are solved using several finite difference schemes. The considerably different solutions from different schemes points out some of their inadequacies.

EQUATION OF FLOW

The differential equation which describes water movement through soil is obtained by substituting Darcy's law into the differential forms of the conservation of mass equation. The following simplified form of this equation will be used

$$\nabla \cdot (K \nabla h) = \eta \frac{\partial S}{\partial t} \quad . \; . \; . \; . \; . \; . \; . \; . \quad (1)$$

in which the hydraulic head h is assumed to consist of the sum of the elevation and pressure heads, K is the hydraulic conductivity with dimensions of velocity, η is the soil porosity, i.e. volume of voids divided by the total volume, S is the moisture saturation, i.e. volume of water divided by the volume of voids, and t is time. The following assumptions are required to arrive at Eq. 1.

1. Darcy's law is valid for saturated and unsaturated flow.
2. The gas flow occurs under such small gradients compared to the water flow that it can be ignored.
3. The fluid (water) is incompressible and consequently of constant density.
4. The solid particles do not move or consolidate, and consequently η is constant.
5. On a macro scale, the functions which describe the flow, and their derivatives, are continuous so that the differential form of the continuity equation is valid.

The hydraulic conductivity K will be expressed as the product of the saturated hydraulic conductivity K_o, which is constant for homogeneous soils, but a scalar variable in space for heterogeneous soil, and the relative hydraulic conductivity K_r which will be assumed a function of the capillary pressure head $p_h = \frac{p}{\rho g}$ for unsaturated conditions and equal to unity for saturated conditions. The discussion will be restricted to homogeneous soils.

Furthermore, the required functional relationships of K_r and S to p_h, which are needed to solve the flow equation, will be defined using the Brooks-Corey equations (Brooks and Corey, 1966) primarily because their use permits the analysis of the flow equation to be more concise, but also because of the simplicity of these equations. Granted for infiltration problems which deal with the imbibition cycle better fits of experimental data can be obtained by other parametric relationships such as those proposed by Brutsaert (1968) and

King (1965). The Brooks-Corey equations are based on only three parameters, the residual saturation S_r, the bubbling pressure, p_b and the pore size distribution exponent λ. Since this report developed from a program with the objective of obtaining parameters which describe hydraulic properties of soils from field infiltrometer tests, it was desirable to keep the number of such parameters to a minimum. Considering the simplicity of the Brooks-Corey equations and how well they fit even imbibition data except in the range approaching unit saturation they were adopted.

From considerations of similitude (Corey and Corey, 1967, and Brooks, et al., 1971), the space coordinates used in defining the problems, the hydraulic head, and the negative of the pressure head will be nondimensionalized by dividing by the bubbling pressure head. In doing this for infiltration problems it is assumed that the bubbling pressure is properly defined as a positive quantity greater than zero for imbibition. Consequently, the physical interpretation of bubbling pressure, often used for drainage data, as the maximum magnitude of negative pressure in the water at which the soil still remains at unit saturation (air entry pressure) is not applicable. Rather p_b is that quantity which gives the best fit of the Brooks-Corey equations to imbibition data. In terms of the dimensionless pressure $p_t = -p/p_b$, the Brooks-Corey equations are:

$$S_e = \frac{S-S_r}{1-S_r} = \frac{1}{p_t^{\lambda}} \quad . \; . \; . \; . \; . \; . \; . \; . \quad (2)$$

and

$$K_r = \frac{1}{p_t^{2+3\lambda}} \quad . \; . \; . \; . \; . \; . \; . \; . \quad (3)$$

There are numerical advantages to applying a transformation to the flow equation which eliminates the square of first derivatives which occur in the expanded form of Eq. 1. Such a transformation of dependent variables is:

$$\xi = \int_{1}^{p_t} K_r(p_t') \, dp_t' \quad . \; . \; . \; . \; . \; . \; . \; . \quad (4)$$

This transformation will be called the Kirchhoff transformation in accordance with Ames (1965) and a number of other authors, even though this transformation or kindred transformations have been denoted otherwise in some literature. Raats and Gardner (1971) and others refer to ξ as the matric flux potential. If K_r is defined by Eq. 3, then Eq. 4 gives the following relations between ξ and p_t.

$$\frac{1}{p_t} = [1 - (1+3\lambda)\xi]^{\frac{1}{1+3\lambda}} \quad . \; . \; . \; . \; . \; . \; . \quad (5)$$

Substituting Eqs. 2 through 5 into Eq. 1 and expanding in cylindrical coordinates for axisymmetric flows gives,

$$\frac{\partial^2 \xi}{\partial r^2} + \frac{\partial^2 \xi}{\partial z^2} + (2+3\lambda)\,[1-(1+3\lambda)\,\xi]^{\frac{1}{1+3\lambda}} \frac{\partial \xi}{\partial z} + \frac{1}{r}\frac{\partial \xi}{\partial r}$$

$$= \frac{1}{[1-(1+3\lambda)\,\xi\,]^{\frac{1+2\lambda}{1+3\lambda}}} \frac{\partial \xi}{\partial \tau} \quad . \; . \; . \; . \; . \; . \; . \quad (6)$$

in which r is the dimensionless radial coordinate in the horizontal plane, z is the dimensionless vertical coordinate, and the dimensionless time parameter τ is given by

$$\tau = \frac{K_o t}{\lambda(p_b/\rho g)\eta(1-S_r)} \quad . \; . \; . \; . \; . \; . \; . \; . \; . \quad (7)$$

The quantity $\eta(1 - S_r)$ in Eq. 7 is often denoted as the effective porosity.

For one-dimensional vertical moisture movement, Eq. 6 reduces to:

$$\frac{\partial^2 \xi}{\partial z^2} + (2+3\lambda)\,[1 - (1+3\lambda)\,\xi\,]^{\frac{1}{1+3\lambda}} \frac{\partial \xi}{\partial z} = \frac{1}{[1 - (1+3\lambda)\,\xi\,]^{\frac{1+2\lambda}{1+3\lambda}}} \frac{\partial \xi}{\partial \tau}$$

$$. \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (8)$$

When associated with appropriate initial and boundary conditions, Eqs. 6 and 8 mathematically describe problems of axisymmetric and vertical moisture movement respectively. For the infiltration problems considered herein the initial condition will assume that static equilibrium exists; that is, at the initiation of the numerical solution no moisture movement occurs anywhere within the region of the problem. From Darcy's law, the hydraulic head is constant under static equilibrium, and consequently, from Eq. 5 and the definition of hydraulic head the initial condition is:

$$\xi(r, z, 0) = [1 - 1/(z - h_o)^{1+3\lambda}] / (1 + 3\lambda) \quad . \quad . \quad . \quad . \quad . \quad . \quad . \qquad (9)$$

in which h_o is the value of the dimensionless constant hydraulic head $p/(\rho g p_b) + z$, which will be specified. Note that the right side of Eq. 9 does not depend upon r and consequently Eq. 9 defines the initial condition for one-dimensional vertical as well as axisymmetric moisture movement.

Boundary conditions for the axisymmetric infiltration problem consisting of moisture applied over a circle of dimensionless radius r_a and moving through a soil of dimensionless depth $D = \text{Depth}/(p_b/\rho g)$ which is underlain by an impervious layer are:

A. A top surface of moisture application

1. Flux rate, $v(t)$ specified,

$$\frac{\partial \xi(r, D, \tau)}{\partial z} = K_r - \frac{|v(\tau)|}{K_o}, \quad 0 < r \le r_a \quad . \quad . \quad . \qquad (10)$$

in which $|v(\tau)|$ is flux per unit area (with dimensions of velocity) and is positive when directed downward in the negative z direction.

2. Surface saturation $S(D, \tau)$ specified (Dirichlet condition)

$$\xi(r, D, \tau) = \left[1 - \left(\frac{S(D,\tau) - S_r}{1 - S_r} \right)^{\frac{1+3\lambda}{\lambda}} \right] \Bigg/ (1 + 3\lambda) \quad . \quad . \qquad (11)$$

B. Top surface beyond radius of application

$$\frac{\partial \xi(r, D, \tau)}{\partial z} = K_r, \quad r > r_a \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \qquad (12)$$

C. Impervious layer

$$\frac{\partial \xi(r, 0, \tau)}{\partial z} = K_r \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \qquad (13)$$

D. Axis of symmetry

$$\frac{\partial \xi(0, z, \tau)}{\partial r} = 0 \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \qquad (14)$$

E. Outer boundary beyond radius of influence, r_f (Dirichlet condition)

$$\xi(r_f, z, \tau) = \xi(r, z, 0) \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \qquad (15)$$

For vertical moisture movement, the boundary conditions are identical to (A) and (C) above except that ξ does not vary over r but only with τ.

FINITE DIFFERENCE SOLUTION

Finite difference methods for solving the initial-boundary-value problems associated with parabolic partial differential equations can be classified as either explicit or implicit. An explicit scheme predicts all quantities at each new advanced time step from known values at the current time step or steps, whereas implicit schemes difference the partial differential equations in such a manner as to require that a system of equations including the boundary condition equations must be solved to determine quantities at each new advanced time step. From stability and convergence analysis of commonly used linear explicit methods, which use second order differences of space derivatives at only the current time step, the condition $\Delta\tau \leq \left\{\frac{\Delta x^2 \quad \Delta y^2}{\Delta x^2 + \Delta y^2}\right\}$ generally severely limits the size of the time step which is permitted. Furthermore, for such methods the discretization errors are of $0(\Delta\tau + (\Delta x)^2 + (\Delta y)^2)$. Consequent implicit methods such as the Crank-Nicolson method which have second order discretization errors in time as well as space (at least for linear equations) are often preferred. This study utilizes the Crank-Nicolson method principally, and obtains the solution to the system of nonlinear algebraic equations therefrom, by the general Newton-Raphson method. Brutsaert (1971) used a similar method of solving the nonlinear algebraic equations from a fully implicit scheme with first order discretization error of the time derivative instead of the Crank-Nicolson method.

Finite Difference Operators for Interior Grid Points

The Crank-Nicolson method weights difference approximations of the derivatives with respect to the space coordinates at the current and advanced time steps equally as the derivative with respect to time is approximated by a second order central difference centered midway between these two time steps. Before carrying out this difference scheme, the writer's first inclination was to multiply Eq. 6 through by $\{1 - (1 + 3\lambda)\xi\}^{\frac{1+2\lambda}{1+3\lambda}}$. It appeared that since this quantity is very small for conditions near static equilibrium, multiplying through by it would minimize truncation errors resulting from division by it. Multiplying through by this quantity causes other difficulties however, as described later. Multiplying through by this quantity and then differencing the space derivatives with second order central difference approximations for implementation of the Crank-Nicolson method, gives the following finite difference operator from Eq. 6 for a square space grid network with $\triangle r = \triangle z = \triangle s$.

$$F_{ij} = a_1^{k+1}(b_1\xi_{i+1j}^{k+1} + b_2\xi_{i+1j}^{k+1}) + (a_1^{k+1} + a_2^{k+1})\,\xi_{ij-1}^{k+1}$$

$$+ (a_1^{k+1} - a_2^{k+1})\,\xi_{ij+1}^{k+1} - (4a_1^{k+1} + c)\,\xi_{ij}^{k+1} + a_1^k(b_1\xi_{i+1j}^k + b_2\xi_{i-1j}^k)$$

$$+ (a_1^k + a_2^k)\,\xi_{ij-1}^k + (a_1^k - a_2^k)\,\xi_{ij+1}^k + (c - 4a_1^k)\,\xi_{ij}^k = 0,$$

$$\text{for} \quad \begin{matrix} i = \\ j = \end{matrix}\begin{Bmatrix} 2, 3 \ldots N_r - 1 \\ 2, 3 \ldots N_z - 1 \end{Bmatrix} \qquad \ldots \ldots \ldots \ldots \quad (16)$$

in which the subscripts i and j denote the space grid points with $i = 1 + r/\triangle s$ and $j = 1 + (D - z)/\triangle s$, the superscript k denotes the time step such that $k = 1 + \tau/\triangle\tau$.

$$a_1 = [1 - (1+3\lambda)\xi_{ij}]^{\frac{1+2\lambda}{1+3\lambda}} \qquad \ldots \ldots \ldots \quad (17)$$

$$a_2 = \frac{(2 + 3\lambda)\triangle s}{2}\,[1 - (1 + 3\lambda)\,\xi_{ij}]^{\frac{1+2\lambda}{1+3\lambda}} \qquad \ldots \quad (18)$$

(In Eqs. 17 and 18, the superscripts k of ξ corresponds to the superscript of a in Eq. 16.)

$$c = \frac{2\triangle s^2}{\triangle\,\tau} \qquad \ldots \ldots \ldots \ldots \quad (19)$$

$$b_1 = 1 + \frac{\triangle s}{2r} = 1 + \frac{.5}{i-1} \qquad \ldots \ldots \ldots \quad (20)$$

$$b_2 = 1 - \frac{\triangle s}{2r} = 1 - \frac{.5}{i-1} \qquad \ldots \ldots \ldots \quad (21)$$

If Eq. 6 is differenced without prior multiplication by a_1, either the operator G_{ij} or H_{ij}, which are given below, result depending respectively upon whether the coefficient of the time derivative term $\partial\xi/\partial\tau$ is approximated using the average value of ξ_{ij}^{k+1} and ξ_{ij}^k, or approximated by evaluating the coefficient which is associated with ξ_{ij}^{k+1} of the time difference at the advanced time step $k+1$ with ξ_{ij}^{k+1} as the argument and the coefficient which is associated with ξ_{ij}^{k} of the time difference from ξ_{ij}^{k}. (As will be demonstrated later, the solution obtained by use of these two operators is considerably different despite the fact that if the coefficient were constant, they would be identical, and that the latter is invalid being an approximation of $\partial(a_4\xi)/\partial\tau$.)

$$G_{ij} = b_1 \xi_{i+1j}^{k+1} + b_2 \xi_{i+j}^{k+1} + (1+a_3^{k+1}) \xi_{ij-1}^{k+1} + (1-a_3^{k+1}) \xi_{ij+1}^{k+1}$$

$$- 4\xi_{ij}^{k+1} - a_4^{k+1/2}(\xi_{ij}^{k+1} - \xi_{ij}^{k}) + b_1 \xi_{i+1j}^{k} + b_2 \xi_{i-1j}^{k}$$

$$+ (1 + a_3^k) \xi_{ij-1}^{k} + (1 - a_3^k) \xi_{ij+1}^{k} - 4 \xi_{ij}^{k} = 0 \quad . \; . \; . \quad (22)$$

and

$$H_{ij} = b_1 \xi_{i+1j}^{k+1} + b_2 \xi_{i-1j}^{k+1} + (1 + a_3^{k+1}) \xi_{ij-1}^{k+1} + (1 - a_3^{k+1}) \xi_{ij+1}^{k+1}$$

$$- (4 + a_5^{k+1}) \xi_{ij}^{k+1} + b_1 \xi_{i+1j}^{k} + b_2 \xi_{i-1j}^{k} + (1 + a_3^k) \xi_{ij-1}^{k}$$

$$+ (1 - a_3^k) \xi_{ij+1}^{k} + (a_5^k - 4) \xi_{ij}^{k} = 0 \quad . \; . \; . \; . \; . \; . \quad (23)$$

in which

$$a_3 = \Delta s\,(1 + 1.5\lambda)\,[1 - (1 + 3\lambda)\xi_{ij}]^{\frac{1}{1+3\lambda}} \quad . \; . \; . \quad (24)$$

$$a_5 = \frac{c_1}{a_1} = \frac{2\Delta s^2}{\Delta\tau\,[1 - (1 + 3\lambda)\xi_{ij}]^{\frac{1+2\lambda}{1+3\lambda}}} \quad . \; . \; . \quad (25)$$

(The superscripts of ξ in Eqs. 24 and 25 correspond to those in a_3 and a_4 in Eqs. 22 and 23.) And

$$a_4^{k+1/2} = \frac{2\Delta s^2}{\Delta\tau\,[1-(.5+1.5\lambda)\,(\xi_{ij}^{k+1} + \xi_{ij}^{k})]^{\frac{1+2\lambda}{1+3\lambda}}} \quad . \; . \quad (26)$$

For one-dimensional vertical moisture movement, the finite difference operators F, G and H from Eq. 8 are:

$$F_j = (a_1^{k+1} + a_2^{k+1})\xi_{j-1}^{k+1} + (a_1^{k+1} - a_2^{k+1})\xi_{j+1}^{k+1} - (2a_1^{k+1} + c)\xi_j^{k+1} + (a_1^k + a_2^k)\xi_{j-1}^k$$

$$+ (a_1^k - a_2^k)\xi_{j+1}^k + (c - 2a_1^k)\xi_j^k = 0 \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (27)$$

$$G_j = (1 + a_3^{k+1})\xi_{j-1}^{k+1} + (1 - a_3^{k+1})\xi_{j+1}^{k+1} - 2\xi_j^{k+1} - a_4^{k+\frac{1}{2}}(\xi_j^{k+1} - \xi_j^k) + (1 + a_3^k)\xi_{j-1}^k$$

$$+ (1 - a_3^k)\xi_{j+1}^k - 2\xi_j^k = 0 \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (28)$$

$$H_j = (1 + a_3^{k+1})\xi_{j-1}^{k+1} + (1 - a_3^{k+1})\xi_{j+1}^{k+1} - (2 + a_5^{k+1})\xi_j^{k+1} + (1 + a_3^k)\xi_{j-1}^k$$

$$+ (1 - a_3^k)\xi_{j+1}^k + (a_5^k - 2)\xi_j^k = 0 \quad . \; . \; . \; . \; . \; . \; . \; . \quad (29)$$

Finite Difference Operators for Boundary Grid Points

The finite difference equations for the top surface boundary (when not of the Dirichlet type) and the impervious layer boundary have been obtained by approximating the derivatives in the boundary conditions with second order central differences centered on the boundary, and subsequently eliminating the value of ξ at the nonexistent grid point outside the boundary by combining with the appropriate finite difference operator for interior grid points. This procedure gives the following operators for the top surface boundary from Eqs. 10, 12 and 13 for the axisymmetric problem:

$$F_{i1} = a_1^{k+1}(b_1\xi_{i+1,1}^{k+1} + b_2\xi_{i-1,1}^{k+1}) + 2a_1^{k+1}\xi_{i2}^{k+1} + 2\Delta s\,(a_1^{k+1} + a_2^{k+1})(K_r^{k+1} - v_1^{k+1})$$

$$- (4a_1^{k+1} + c)\xi_{i1}^{k+1} + a_1^k(b_1\xi_{i+1,1}^k + b_2\xi_{i-1,1}^k) + 2a_1^k\xi_{i2}^k + 2\Delta s(a_1^k + a_2^k)(K_r^k - v_1^k)$$

$$+ (c - 4a_1^k)\xi_{i1}^k = 0 \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (30)$$

$$G_{i1} = b_1\xi_{i+1,1}^{k+1} + b_2\xi_{i-1,1}^{k+1} + 2\xi_{i2}^{k+1} + 2\Delta s(1 + a_3^{k+1})(K_r^{k+1} - v_1^{k+1}) - 4\xi_{i1}^{k+1}$$

$$- a_4^{k+\frac{1}{2}}(\xi_{i1}^{k+1} - \xi_{i1}^k) + b_1\xi_{i+1,1}^k + b_2\xi_{i-1,1}^k + 2\xi_{i2}^k + 2\Delta s(1 + a_3^k)(K_r^k - v_1^k)$$

$$- 4\xi_{i1}^k = 0 \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (31)$$

$$H_{i1} = b_1\xi_{i+1,1}^{k+1} + b_2\xi_{i-1,1}^{k+1} + 2\xi_{i2}^{k+1} + 2\Delta s(1+a_3^{k+1})(K_r^{k+1} - v_1^{k+1}) - (4+a_5^{k+1})\xi_{i1}^{k+1}$$

$$+ b_1\xi_{i+1,1}^{k} + b_2\xi_{i-1,1}^{k} + 2\xi_{i2}^{k} + 2\Delta s(1+a_3^{k})(K_r^{k} - v_1^{k}) + (a_5^{k} - 4)\xi_{i1}^{k} = 0$$

$$\quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (32)$$

for $(i = 2, 3 \ldots N_r - 1)$ and $v_1 = |v(\tau)| / K_o$ for $i \le (1 + r_a/\Delta s)$ and $v_1 = 0$ for $i > (1 + r_a/\Delta s)$.

The finite difference equations giving F_1, G_1 and H_1 for the one-dimensional vertical moisture movement problem can be obtained from Eqs. 30, 31 and 32 respectively by deleting the terms involving $\xi_{i+1,1}$ and $\xi_{i-1,1}$ and replacing the 4 by a 2 in the quantity which multiplies $\xi_{i,1}$.

The finite difference equations for the bottom impervious layer boundary where $j = N_z$ are quite similar to Eqs. 30 through 32 with $v_1 = 0$, ξ_{i,N_z-1} replacing $\xi_{i,2}$ and the term multiplying K_r changed to $2\Delta s(a_3 - 1)$. These equations will not be given in entirety but for illustrative purposes,

$$G_{iN_z} = b_1\xi_{i+1N_z}^{k+1} + b_2\xi_{i-1N_z}^{k+1} + 2\xi_{iN_z-1}^{k+1} + 2\Delta s(a_3^{k+1} - 1)K_r^{k+1} - 4\xi_{iN_z}^{k+1}$$

$$- a_4^{k+\frac{1}{2}}(\xi_{iN_z}^{k+1} - \xi_{iN_z}^{k}) + b_1\xi_{i+1N_z}^{k} + b_2\xi_{i-1N_z}^{k} + 2\xi_{iN_z-1}^{k} + 2\Delta s(a_3^{k} - 1)K_r^{k}$$

$$- 4\xi_{iN_z}^{k} = 0 \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (33)$$

For the one-dimensional problem, only the above two boundary conditions exist. For the axisymmetric problem the boundary condition at the axis of symmetry where $r = 0$ has been handled by setting $\xi_{1,j} = \xi_{2,j}$ in all the above finite difference equations. This approach was used instead of developing an operator by combining the central difference approximation of Eq. 14 with the regular operator as was done to develop the above equations because b_1 and b_2 are undefined along the line singularity at $r = 0$. At the outer boundary of the region of interest no finite operator is needed since this is a Dirichlet condition.

In order to perform the calculations, which are needed to advance a solution through each time step as described later, only at space grid points within the region of moisture movement and not far beyond the wetting front special finite difference operations have been used to set up artificial boundaries within the region of the defined problem. At such an artificial boundary where $i = M_r$, this special operator is identical to the regular operator F, G or H with the exception that $\xi^{k+1}_{M_r+1\,j} = \xi^{k}_{M_r+1\,j}$. At the artificial boundary where $j = M_z$, the special operator is obtained by setting $\xi^{k+1}_{i\,M_z+1} = \xi^{k}_{i\,M_z+1}$ in Eqs. 16, 22 or 23.

Method of Solving Finite Difference Equations

When the finite difference operators (interior and boundary) are written for all grid points simultaneously a system of nonlinear algebraic equations results for the unknown ξ^{k+1}_{ij}. (The ξ with a k superscript is known.) This system is nonlinear since the a's are functions of ξ, and the number of these equations equals the number of grid points within the region of computation. The solution of this system advances the solution of the infiltration problem through one time step $\Delta\tau$.

The general Newton-Raphson iterative method has been utilized in solving this system of nonlinear algebraic equations. This method provides a better approximation to the unknown vector $\vec{\xi}^{\,k+1} = \xi_{1,1}, \xi_{2,1} \cdots \xi_{M_r,1}, \xi_{2,1} \cdots \xi_{M_r M_z}$ after each new iteration by means of the formula (see Saaty and Bram (1964) for example).

$$(\vec{\xi}^{\,k+1})^{m+1} = (\vec{\xi}^{\,k+1})^{m} - (\mathbf{D}^{-1})^{m}\,(\vec{F})^{m} \quad . \quad . \quad . \quad (34)$$

in which the superscript m outside the parentheses denotes the iteration number, the vector $\vec{F}$ consists of the elements composed of the finite difference operators F_{ij}, and G_{ij} or H_{ij} depending upon the finite difference approximation used, each of which will equal zero when the solution has been obtained, and $\mathbf{D}$ is the Jacobian which for the two-dimensional axisymmetric problem consists of the banded matrix,

$$\mathbf{D} = \left|\begin{array}{cccccccccc}
\frac{\partial F_{11}}{\partial \xi_{11}} & \frac{\partial F_{11}}{\partial \xi_{21}} & 0 \cdots & & 0 & \frac{\partial F_{11}}{\partial \xi_{12}} & 0 \cdots & & & \\
\frac{\partial F_{21}}{\partial \xi_{11}} & \frac{\partial F_{21}}{\partial \xi_{21}} & \frac{\partial F_{21}}{\partial \xi_{31}} & 0 \cdots & 0 & & \frac{\partial F_{21}}{\partial \xi_{22}} & 0 \cdots & & \\
0 & \frac{\partial F_{31}}{\partial \xi_{21}} & \frac{\partial F_{31}}{\partial \xi_{31}} & \frac{\partial F_{31}}{\partial \xi_{41}} & 0 \cdots & 0 & & \frac{\partial F_{31}}{\partial \xi_{32}} & 0 \cdots & \\
 & \ddots & \ddots & \ddots & & & & \ddots & & \\
\frac{\partial F_{12}}{\partial \xi_{11}} & 0 \cdots 0 & \frac{\partial F_{12}}{\partial \xi_{12}} & \frac{\partial F_{12}}{\partial \xi_{22}} & \cdots & & & \frac{\partial F_{12}}{\partial \xi_{13}} & & \\
0 & \ddots\, \frac{\partial F_{22}}{\partial \xi_{21}} & & & \ddots & & & & & \\
 & & & \frac{\partial F_{N_r N_z}}{\partial \xi_{N_r N_z - 1}} & \cdots & & & \frac{\partial F_{N_r N_z}}{\partial \xi_{N_r - 1\, N_z}} & \frac{\partial F_{N_r N_z}}{\partial \xi_{N_r N_z}} &
\end{array}\right| \quad . \quad . \quad (35)$$

The actual implementation of Eq. 34 in a computer program will take the form

$$(\vec{\xi}^{\,k+1})^{m+1} = (\vec{\xi}^{\,k+1})^{m} - \vec{X} \quad \ldots\ldots\ldots\ldots \quad (36)$$

in which the vector $\vec{X}$ is the solution to the linear system

$$(\mathbf{D})^{m}\vec{X} = (\vec{F})^{m} \quad \ldots\ldots\ldots\ldots \quad (37)$$

The equations needed for evaluating the individual derivatives in the matrix $\mathbf{D}$ are given below only for the finite difference operator G_{ij},

$$\frac{\partial G_{i1}}{\partial \xi_{i-1,1}} = b_2 \quad \ldots\ldots\ldots\ldots \quad (38)$$

$$\frac{\partial G_{i1}}{\partial \xi_{i1}} = 2\Delta s\ [(v_1^{k+1} - K_r^{k+1})a_3^{k+1} - (1+a_3^{k+1})(2+3\lambda)K_r^{k+1}]/[1-(1+3\lambda)\xi_{ij}^{k+1}]$$

$$- a_4^{k+\frac{1}{2}}(.5+1.0\lambda)(\xi_{ij}^{k+1} - \xi_{ij}^{k})/[1-(.5+1.5\lambda)(\xi_{ij}^{k+1}+\xi_{ij}^{k})]$$

$$- a_4^{k+\frac{1}{2}} - 4 \quad \ldots\ldots\ldots\ldots \quad (39)$$

$$\frac{\partial G_{i1}}{\partial \xi_{i+1,1}} = \frac{\partial G_{ij}}{\partial \xi_{i+1,1}} = \frac{\partial G_{iN_z}}{\partial \xi_{i+1,1}} = b_1 \quad \ldots\ldots\ldots\ldots \quad (40)$$

$$\frac{\partial G_{i1}}{\partial \xi_{i2}} = \frac{\partial G_{iN_z}}{\partial \xi_{iN_z-1}} = 2 \quad \ldots\ldots\ldots\ldots \quad (41)$$

$$\frac{\partial G_{i1}}{\partial \xi_{i-1,1}} = \frac{\partial G_{ij}}{\partial \xi_{i-1j}} = \frac{\partial G_{iN_z}}{\partial \xi_{i-1N_z}} = b_2 \quad \ldots\ldots\ldots\ldots \quad (42)$$

$$\frac{\partial G_{ij}}{\partial \xi_{ij}} = a_3^{k+1}(\xi_{ij+1}^{k+1} - \xi_{ij-1}^{k+1})/[1-(1+3\lambda)\xi_{ij}^{k}] - \frac{a_4^{k+\frac{1}{2}}(.5+\lambda)(\xi_{ij}^{k+1}-\xi_{ij}^{k})}{1-(.5+1.5\lambda)(\xi_{ij}^{k+1}+\xi_{ij}^{k})}$$

$$- a_4^{k+\frac{1}{2}} - 4 \quad \ldots\ldots\ldots\ldots \quad (43)$$

$$\frac{\partial G_{ij}}{\partial \xi_{ij-1}} = 1 + a_3^{k+1} \quad \ldots\ldots\ldots\ldots \quad (44)$$

$$\frac{\partial G_{ij}}{\partial \xi_{ij+1}} = 1 - a_3^{k+1} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (45)$$

$$\frac{\partial G_{i N_z}}{\partial \xi_{i N_z}} = 2\Delta s \; [\{(1 - a_3^{k+1})(2+3\lambda) - a_3^{k+1}\} K_r^{k+1}]/[1\,(1+3\lambda)\xi_{ij}^{k+1}]$$

$$- a_4^{k+\frac{1}{2}}(.5+\lambda)(\xi_{ij}^{k+1} - \xi_{ij}^{k})/[1 - (.5 + 1.5\lambda)(\xi_{ij}^{k+1} + \xi_{ij}^{k})] - a_4^{k+\frac{1}{2}} - 4$$

$$. \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (46)$$

For the one-dimensional problem the derivative in **D** can be obtained from Eqs. 38 through 46 by eliminating terms involving ξ_{i+1j} and ξ_{i-1j}. Furthermore, for the one-dimensional problem, **D** reduces to a tridiagonal matrix, which from a computational point of view is important since a system with a tridiagonal coefficient matrix can be solved efficiently by the Thomas algorithm (a triangularization scheme given in a number of sources (see Remson et al., 1971), or a one pass elimination of the elements below the diagonal followed by back substitution).

For the two-dimensional problem the solution vector $\vec{X}$ was initially obtained by using the algorithm described by Thurman (1963). (The actual listing of the ALGOL program was obtained from Burrough Corporation.) This algorithm is more efficient in computation and storage requirements than standard linear algebra algorithms because it operates only on the band portion of the matrix **D**. Subsequently, upon noting that the diagonal elements of **D** are considerably larger in magnitude than the off-diagonal nonzero elements, particularly during the first time steps of the solution, it became apparent that considerable reduction in storage requirements could be achieved and at the same time possibly decrease the amount of computer execution time required for a solution, by utilizing an inner iterative scheme. This scheme in essence combines the line successive relaxation iterative method with the Newton-Raphson method; that is an iteration is created within an iteration. The inner most iteration solves for the ξ^{k+1}'s along consecutive horizontal lines from the system of equations resulting under the assumption that the ξ^{k+1}'s on the previous and the next line are known. In other words, the two outer bands of **D** in Eq. 35 were assumed zero and consequently a tridiagonal system of equations, equal in number to the grid points along this line, is solved during each inner iteration. This inner iteration is continued until the sum of absolute changes in ξ's along the line becomes less in magnitude than a specified error (i.e approximately 10^{-7}), before repeating the same inner iterative process at the next line. A pass through all lines constitutes one outer iteration, and provides values of ξ throughout the flow region which are close to those that would be obtained from one iteration by the Newton-Raphson method, Eq. 36. During each outer iteration the sum of accumulated absolute changes in ξ from all of the inner iterations along individual lines is accumulated. When this sum becomes less than a second error parameter, the iteration is terminated.

During initial time steps, generally no more than three inner iterations and four outer iterations are required to satisfy error parameter of 10^{-7}. As the moisture movement penetrates to depth of one or more units, the number of inner iterations increases to perhaps as many as 10 for lines in the vicinity of the wetting front and as many as 20 outer iterations may be required to satisfy an error parameter of 10^{-7}. The above solution processes will be referred to as the Newton-Line-Relaxation method. Over relaxation, or over adjustment, of individual lines has not been studied.

The logic required to program the Newton-Line-Relaxation method using a space grid network which continually expands in both the r and z directions is straightforward and considerably simplier than the logic required to program the Newton-Raphson method without the inner iteration. Point by point iterative methods such as Lieberstein's (1959) method of nonlinear simultaneous displacements or his nonlinear generalization of the SOR method likely

$$\xi_{ij}^{(m+1)} = \xi_{ij}^{(m)} - w \frac{F_{ij}(\xi_{11}^{(m+1)}, \ldots \xi_{i-1j-1}^{(m+1)}, \xi_{ij}^{(m)}, \ldots \xi_{M_rM_z}^{(m)})}{\dfrac{\partial F_{ij}(\xi_{11}^{(m+1)}, \ldots \xi_{i-1j-1}^{(m+1)}, \xi_{ij}^{(m)}, \ldots \xi_{M_rM_z}^{(m)})}{\partial \xi_{ij}}} \quad \ldots \quad (47)$$

would also be satisfactory. Bryan (1964) treats the convergence of such point by point iterative methods for solving nonlinear systems of algebraic equations.

The method of developing the finite difference equations does have considerable influence on the magnitude of the numbers which represent the finite difference solution. The magnitude of the a coefficients in the finite difference equations vary considerable with changes in ξ, particularly when no moisture movement exists. Consequently, the method used to evaluate these coefficients is quite critical. The variation of the coefficient a_4 which multiplies the derivative with respect to the time parameter is shown in Fig. 1. The spacing of increments on the ordinate on Fig. 1 has been changed by a factor of 10 between the four portions of the graph separated by horizontal lines. The value of ξ which corresponds to the static equilibrium condition is in the left most region of each curve on Fig. 1 where the curves are very steep. Thus for example with $\lambda = 0.9$ and an initial hydraulic head of -3 units and a depth of 2 units so that the pressure head on the surface equals -5 units, the static equilibrium value of $\xi =$ .2695694 on the surface produces the value for the coefficient $a_4 = 18.1$. A reduction of ξ to .260 causes a_4 to reduce to 2.4.

Common methods of evaluating the coefficients are:

1. From values of the dependent variable at the current time step (see Hanks and Bowers (1962) and Jeppson (1970) for example).

2. By means of a predictor to estimate the dependent variable at the advanced time step and base the coefficient on this estimated value (see Rubin (1968) for example).

3. Use of a Picand type iteration to improve an initial guess (see Klute et al. (1965) and Smith and Woolhiser (1971) for example).

4. Solve the system fully implicitly in which the coefficients are part of the unknown at the advanced time step such as implied by the operators F, G and H given earlier (see Brutsaert, 1971).

The predictor used in method 2 can be obtained by noting that an approximation for ξ^{k+1} at the advanced time step is given by,

$$\xi^{k+1} = \xi^{k} + \left(\frac{\partial \xi}{\partial \tau}\right)^{k} \Delta \tau \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (48)$$

But $(\partial \xi / \partial \tau)^{k}$ can be evaluated by solving for it from Eq. 8 and evaluating the other side of the equal sign by appropriate differences at the current time step. Method 4 requires solving a system of nonlinear algebraic equations[1/] whereas the other methods required the solution of a linear system, even though this may be more than once.

The variation of the coefficients a_4 at the axis of symmetry computed by methods 1, 2 and 4 are shown in Fig. 2 for the beginning few time steps for a two-dimensional problem being solved using the G_{ij}

[1/] Note that use of the Newton-Raphson method actually accomplishes the solution of the nonlinear system by iteratively solving a linear system.

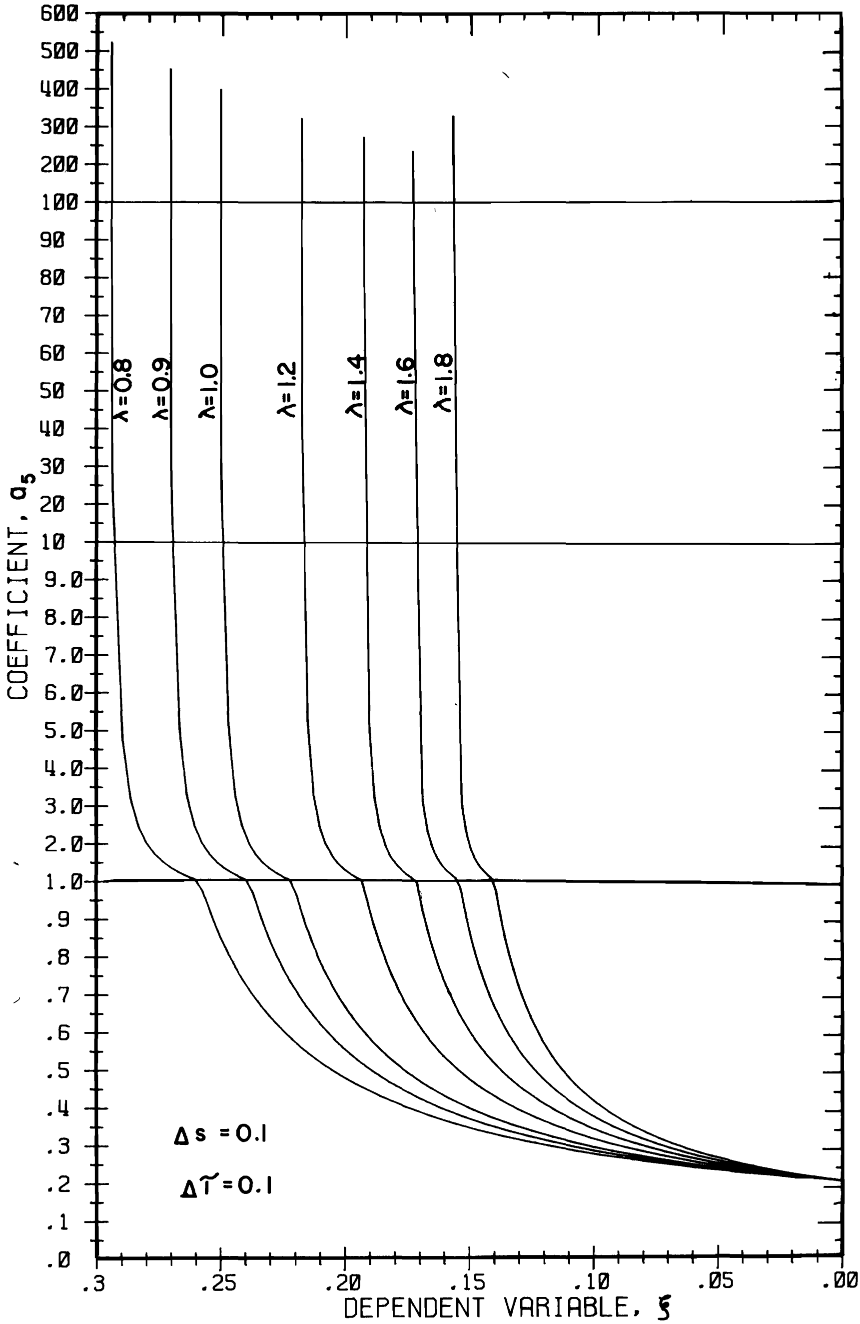


FIG. 1. VARIATION OF THE COEFFICIENT a_5 WITH ξ AND λ.

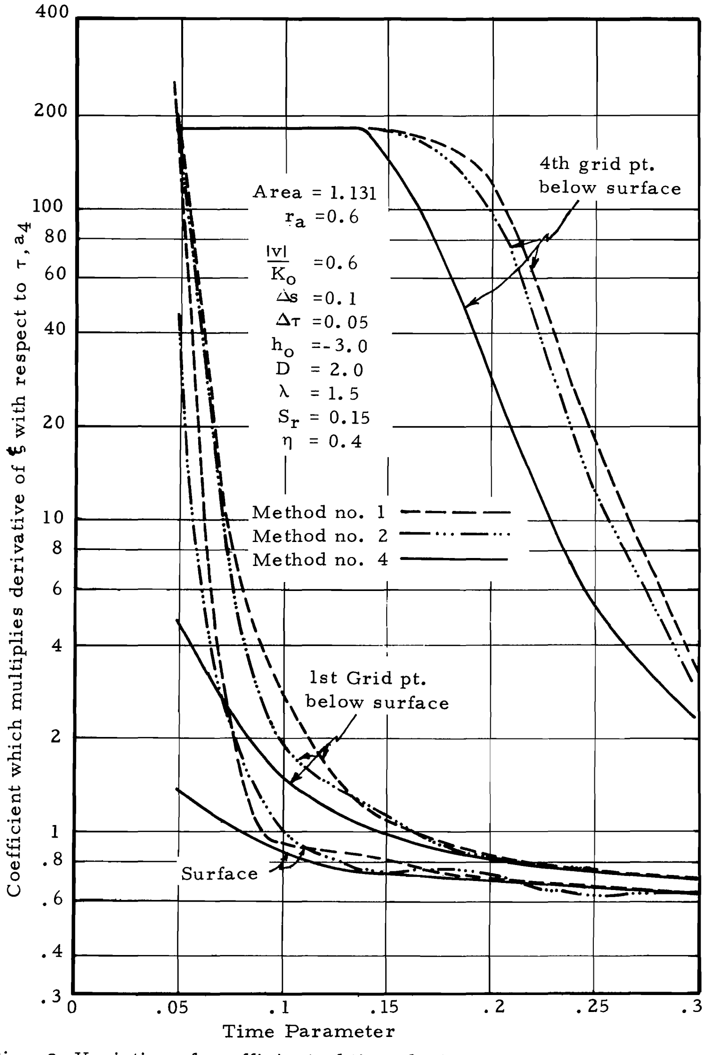


Fig. 2. Variation of coefficient of time derivative at two grid points by three different methods over several time steps in obtaining solution to given problem by the Crank-Nicolson method using finite difference operators, G_{ij}.

operator. The basic problem specifications are given on the graph and each group of three lines on the graph show these coefficients at the finite difference grid point on the surface, the first grid point 0.1 unit below the surface, and the fourth grid point below the surface. This figure shows that in the vicinity of the wetting front methods 1 and 2 considerably over estimate the magnitude of a_4. After the wetting front has passed, i.e. the coefficients approach a constant value, all methods yield coefficients of equal magnitude. The same trend shown in Fig. 2 has been verified by examining the same coefficients at other points at different depths and radii. Near the surface where the wetting front is the sharpest, method 2 supplies coefficients which may oscillate between the correct value and those given by method 1, to above those given by method 1. It follows that if methods 1 and 2 supply values of a_4 which are too large, then these methods supply values of a_1 and a_2 which are too small.

Solutions Obtained by Several Different Finite Difference Equations

In obtaining a solution to a system of nonlinear algebraic equations by the Newton-Raphson method it is necessary to supply an initial guess for the unknown vector, which for the system given herein is $(\vec{\xi}^{\,k+1})^0$. Convergence of the Newton-Raphson iteration will occur only if this initial guess is close enough to the correct solution. The closeness required depends upon the characteristics of the system of equations. For some systems most any rough initial guess may be adequate, whereas other systems may require that the guess be very close. For the infiltration problems described herein it is not difficult to supply a reasonably close initial guess. After all, the values of $\vec{\xi}^{\,k}$ at the current time step would generally be considered reasonably close to $\vec{\xi}^{\,(k+1)}$ at the advanced time step, and keeping track of changes between consecutive time steps allows an even better initial guess to be supplied.

In attempting to obtain a solution to the two-dimensional problem using the F operators, the writer discovered he could not program an adequate initial guess by means of a couple of statements. Even after achieving success in getting a proper solution for a few time steps by meticulously adjusting the initial guesses, the Newton-Raphson iteration failed in solving the system at later time steps. An examination of reasons why what the writer considered a very close initial guess was not adequate is quite revealing. To simplify this examination, the finite difference operators for the one-dimensional vertical moisture movement problem will be examined. The two-dimensional operator has similar characteristics, but more complicated by possible changes in the radial direction.

The functions F_j, $j = 1, 2 \ldots N_z$ actually depend upon the magnitude of ξ_j^{k+1} at all grid points throughout the soil profile. The dependency is greatest, however, on ξ_{j-1}^{k+1}, ξ_j^{k+1}, and ξ_{j+1}^{k+1} because these are the only values appearing in each individual equation for F_j. The influence from the other ξ's is indirect in that their magnitudes effect the magnitudes of these three ξ's. Graphically only the variation of F_j with one ξ can be displayed conveniently in a single plot. Fig. 3 shows how F_1 varies with ξ_1 for six different application rates $Q = |v|/K_o$. All of these curves have been obtained using a value of ξ_2^{k+1} which satisfies the static equilibrium condition to 16 significant digits. The zero for the function F_1 occurs when the curve intersects the horizontal dashed line approximately midway through the graph. In plotting the curves on this figure each curve ends at the right of the graph at a point beyond which the function F_1 becomes undefined because the quantity $\{1 - (1 + 3\lambda)\xi_j^{k+1}\}$ which is raised to an exponent in the equation for F_1 becomes negative. For the smaller application rates it appears that the point where F_1 becomes undefined and where its zero exists are coincident. This is not the case, however, as shown on Figs. 4 and 5 in which the plotting in the immediate vicinity of this point has been greatly magnified, for the curves of $Q = .001$ and $Q = .01$ respectively. Rather the following are true.

1. The zero of F_1 occurs at a value of ξ_1 which is very close to the value where F_1 becomes undefined but which is always less.

2. The function F_1 is positive for ξ_1 less than its value at which $F_1 = 0$ and negative for ξ_1 greater than this value.

3. Between the zero of F_1 and its undefined region, F_1 reaches a minimum. For ξ_1 less than the value causing the minimum $\partial F_1/\partial \xi_1$ is negative and for ξ_1 greater than this value $\partial F_1/\partial \xi_1$ is positive.

4. As the application rate decreases the phenomena described in (1) through (3) occur essentially at the same point, and the separate phenomena can be detected only by examining the values written out to a relatively large number of significant digits.

This erratic variation of F_1 is due to the strong effect that small changes in ξ_1^{k+1} have on the magnitude of the coefficients a_1^{k+1} and a_2^{k+1} in F_1.

This erratic functional behavior is not restricted to the finite difference operator F_1 for the surface grid point, but is characteristic of the F's at

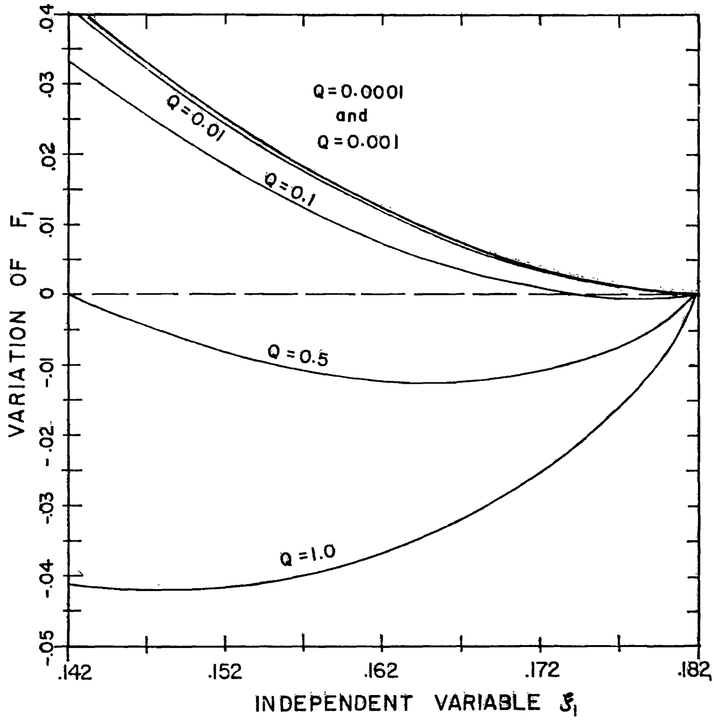


Fig. 3. Functional Relationship of F_1 for several Application rates, Q.

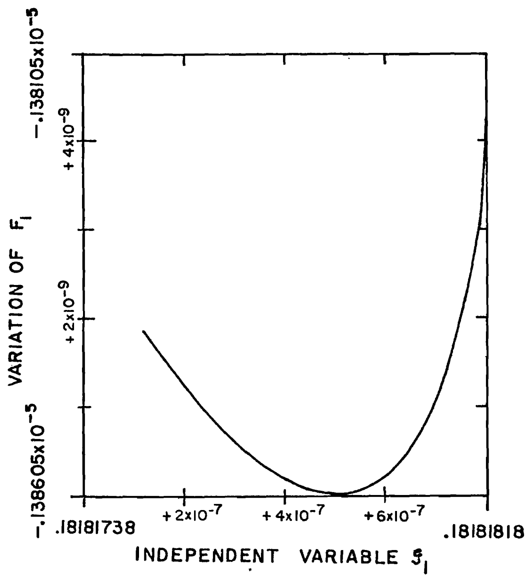


Fig. 4. Functional relationship of F_1 with dimensionless application rate, $Q = 0.001$.

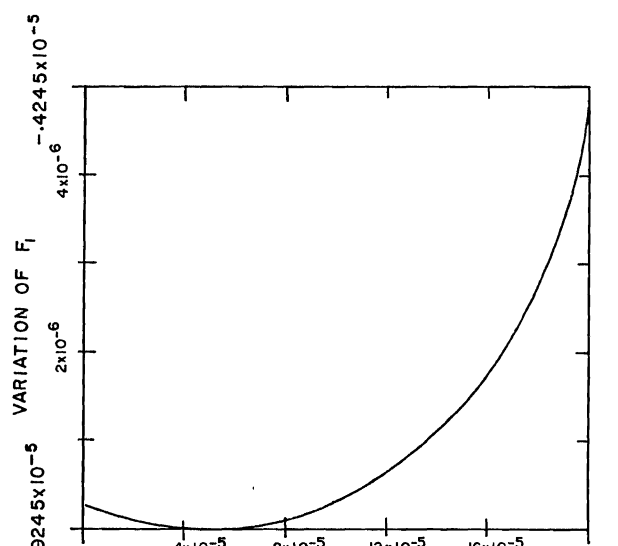


Fig. 5. Functional relationship of F_1 with dimensionless application rate, $Q = 0.01$.

all grid points. The relationship of F_2 at the next grid point with ξ_2 varying over a relatively small range is given in Fig. 6. The separate curves on Fig. 6 were obtained by setting ξ_1^{k+1} equal to a value which is less than its static equilibrium value by the amount indicated on each curve.

It is not difficult to appreciate why the Newton-Raphson method failed to converge during the first attempts at obtaining solutions. Should the guess for ξ_j^{k+1} be larger than that which gives F_j a minimum, subsequent iterations would cause F_j to rapidly, if not immediately, become undefined. On the other hand, if this guess is much smaller than the zero of F_j, the first iteration would project into regions where F_j is undefined. Furthermore, if any iteration should cause F_j to take on its minimum value or close to it, the next vector of $\vec{\xi}_j^{k+1}$ from the Newton-Raphson iteration would be unrealistically large or small. Clearly, to solve the problem using the F operators requires some approach other than the Newton-Raphson method, at least until the values of ξ's are smaller than those associated with no moisture movement.

For a Gauss-Seidel type iteration to converge in solving a system of equations such as those given by the F's a necessary condition is that the ξ being solved for on the left side of the equal sign exert equal, and in at least one equation greater, influence on the function than the composite influence of all other ξ's on the right of the equal sign. For a linear system this condition is referred to as diagonal dominance. From the characteristics of F_j this type of simple iteration would not converge in obtaining even the root ξ_j^{k+1} from a single F_j if ξ_{j+1}^{k+1} and ξ_{j-1}^{k+1} were given correct values because each term in F_j which contains ξ_j^{k+1} exerts dominant influence in a different portion of the domain of interest. Only if the ξ_j^{k+1} placed on the left of the equal sign were changed between iterations depending upon the range of the values, could convergence be achieved.

The approach which has been used to successfully solve problems using the F finite difference operators takes advantage of the fact that $\partial F_j/\partial \xi_j$ in the vicinity of the root is larger than either $\partial F_j/\partial \xi_{j-1}$ or $\partial F_j/\partial \xi_{j+1}$ and therefore if the correct root for each separate F_j can be determined, then a Gauss-Seidel type iteration between equations will converge. The root to each F_j is obtained by squaring and utilizing a Fibonacci search (Wilde and Beighter, 1967) to obtain the minimum of the squared function. A small interval for the search is relatively easy to determine from the characteristics of F_j described earlier. After using the Fibonacci search-iterative scheme for a few iterations, the solution process is turned over to the Newton-Raphson iteration to complete the solution for each time step.

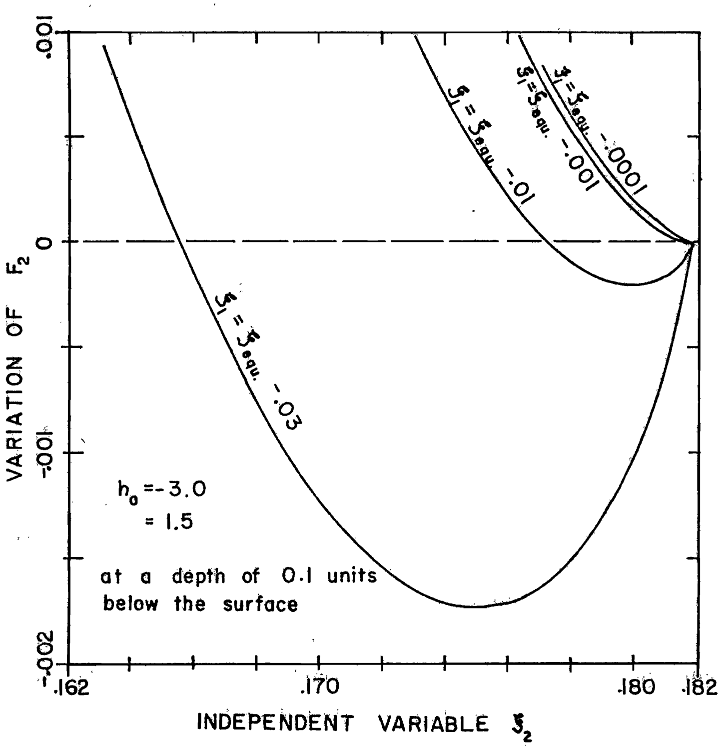


Fig. 6. Functional Relationship of F_2 at the First Grid Point below the Surface with the Value of ξ_2 at the surface grid point decreased by the Indicated Amount.

A much better approach to a solution is available, however, by using the G finite difference operators. Fig. 7 shows that the G_j's do not behave erratically in the vicinity of the root as do the F_j's. Consequently, no particular difficulties occur in applying the Newton-Raphson method or the Newton-Line-Relaxation method in obtaining solutions to either one-dimensional or two-dimensional problems resulting from the G finite difference operators. A solution to a typical axisymmetric problem is given in Fig. 8 in which lines of constant saturation have been plotted in a meridial plane beginning at the axis of symmetry from solutions at 9 different time steps.

Solutions based on operators F and G agree closely. The detectable differences are restricted to the region surrounding the wetting front. For example, solving the one-dimensional problem with $\lambda = 1.5$, $h_o = -3.0$, $S_r = .15$, $\eta = .4$, $D = 3.0$, $|v|/K_o = 0.6$, using $\Delta s = 0.1$, $\Delta\tau = .025$ and an error parameter of 5×10^{-8}, shows a difference of 1.71 percent at the surface grid point when $\tau = .05$, with the solution based on G giving the larger saturation. After advancing through 18 time steps ($\tau = .45$), the saturations at the surface agreed to within 0.07 percent and at the wetting front, which at this time has advanced to a depth of approximately 0.6 units, the maximum difference is 4.51 percent with the solution based on G giving the highest saturations. When $\tau = 1.125$ the two solutions show identical saturations at the surface to the four digits printed, at the 0.6 unit depth the difference is within 0.03 percent, and at the wetting front which at this time has advanced just beyond a depth of 1.4 units, the difference is 3.92 percent. The difference at this same time at a 1.3 unit depth is 1.38 percent. The computer program obtaining the solution based on the F operators used double precision, where the program which solved the problem using the G's used single precision. Some of the above differences might be tracable to the difference in arithmetic precision, but regardless the overall solutions from the two operators are close enough that for all practical purposes they are identical. Since the G functions do not exhibit the erratic behavior that the F's do, clearly the use of the G finite difference operators is preferable.

No means exist for quantitatively defining how close the finite difference solutions converge to the actual solution of the nonlinear initial-boundary-value problem. By comparing solutions to the same problem obtained using different time and space increments, an indication is given whether these increments need to be decreased in size or not. The problem specified by $\lambda = 0.9$, $S_r = 0.22$, $\eta = 0.4$, $h_o = -3.0$, $D = 2.0$ and $S(D, \tau) = 0.85$ was solved thrice by means of the G operators using first $\Delta\tau = .05$ and $\Delta s = 0.1$, second $\Delta\tau =$

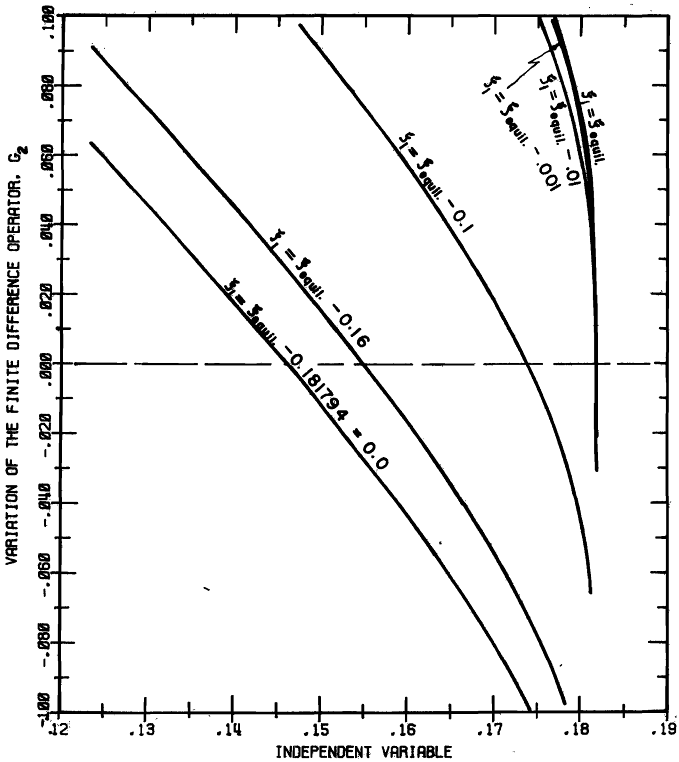


Fig. 7. Functional relationship of G_2 at the first grid point below the surface with ξ_2 and with ξ_1 at the surface decreased by the indicated amount.

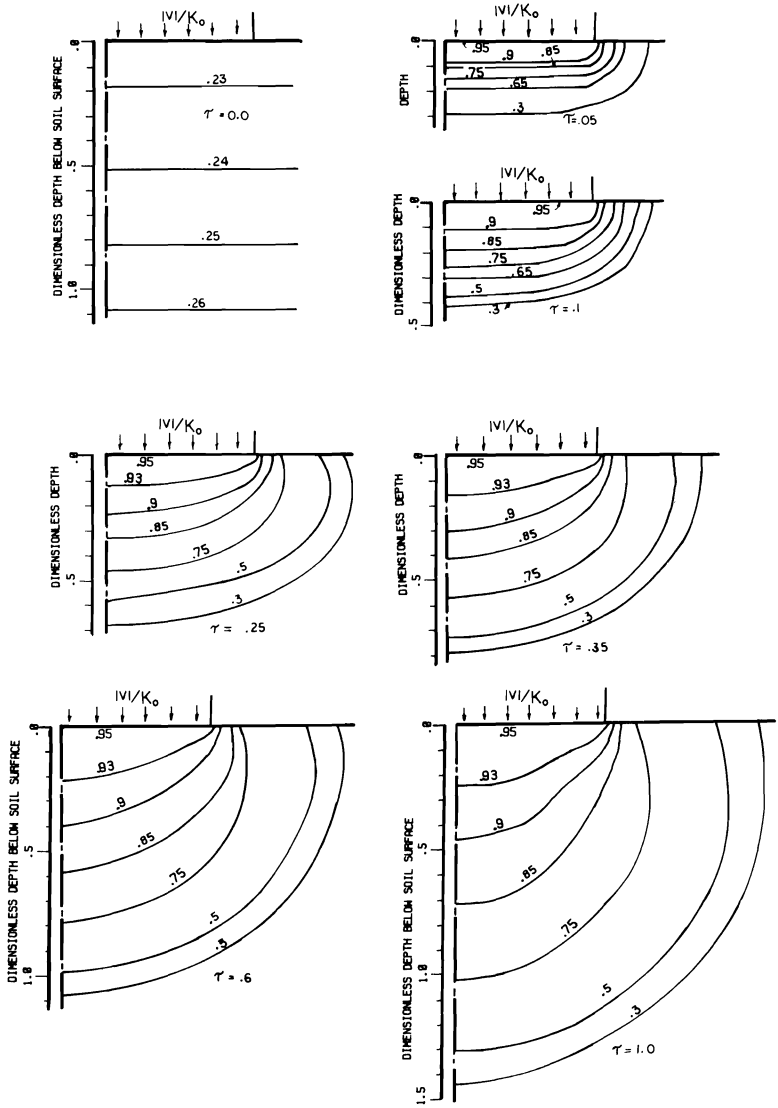
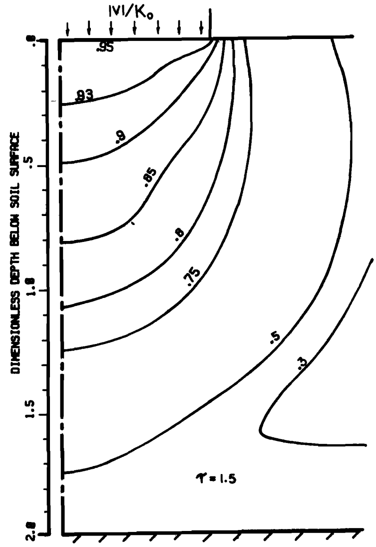


Fig. 8. Saturation distributions at several time steps resulting from solving the problem given by $\lambda = 1.5$, $h_o = -3.0$, $S_r = .15$, $\eta = .4$, $D = 2.0$, $S(1,\tau) = 0.95$.



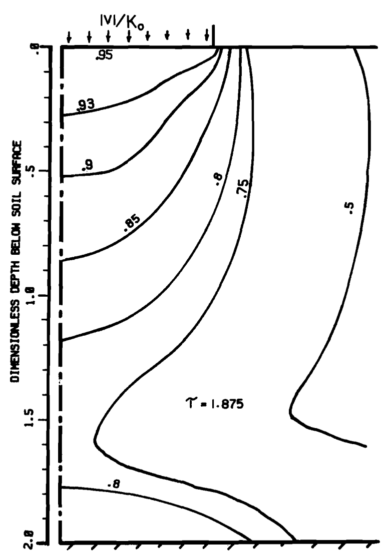


Fig. 8. Continued.

.025 and $\Delta s = 0.1$, and third $\Delta \tau = .025$ and $\Delta s = 0.05$. Very small differences exist between these three solutions. These differences are confined to the vicinity of the wetting front and soon disappear. For instance, the largest difference in saturations between the first and second solution at the time step $\tau = 0.15$ occurs at the first grid point beneath the surface and near the centerline where the difference is 0.41 percent. When $\tau = 0.6$ the difference at this point has diminished to 0.04 percent, but the maximum difference occurs at a depth of 0.7 units and equals 1.5 percent. When $\tau = 1.35$, the surface saturations are identical to at least four digits, the difference at the 0.7 unit depth is 0.12 percent but just behind the wetting front at a depth of 1.0 units the greatest difference equals 0.71 percent. A comparison of the first two solutions with the third shows very small differences, with first and third in closer agreement than the second and third. The results from the first solution lie between those from second and third solutions. For instance, at $\tau = 0.15$ at a depth of 0.1 at the centerline the difference of saturations between the second and third solutions equals 0.62 percent. At the time step when τ = 0.3 this difference is 0.15 percent with the greatest difference of 1.86 percent occurring at a depth of 0.4 units. These small differences lead to the conclusion that reducing the grid spacing or time increment won't significantly improve the finite difference solution.

Of much greater significance, however, is the scheme used in obtaining the finite difference operators. As noted earlier, the F and G finite difference operator produces essentially identical solutions. Solutions based on the H finite difference operators show no quantitative agreement with those obtained from the G operators, however, despite the fact that if the coefficient which multiplies the derivative $\partial\xi/\partial\tau$ were constant the two operators would be identical. For instance, Fig. 9 gives solution results from solving the identical problem using the G and H operators. Also shown on this figure are solution results from solving the same problem with the alternating direction implicit AID method (Douglas, 1961) with a predictor to evaluate the coefficients, as described later. Clearly the H operator is in error. The error is in the method of differencing the right side of Eq. 6. The H operator accomplishes this difference by letting,

$$\frac{\Delta\tau}{\Delta s^2}\, a_5 \frac{\partial\xi}{\partial\tau} \approx (a_5^{k+1}\, \xi_{ij}^{k+1} - a_5^{k}\, \xi_{ij}^{k})/\Delta s^2 \quad . \; . \; . \; . \; . \; . \; . \; . \quad (49)$$

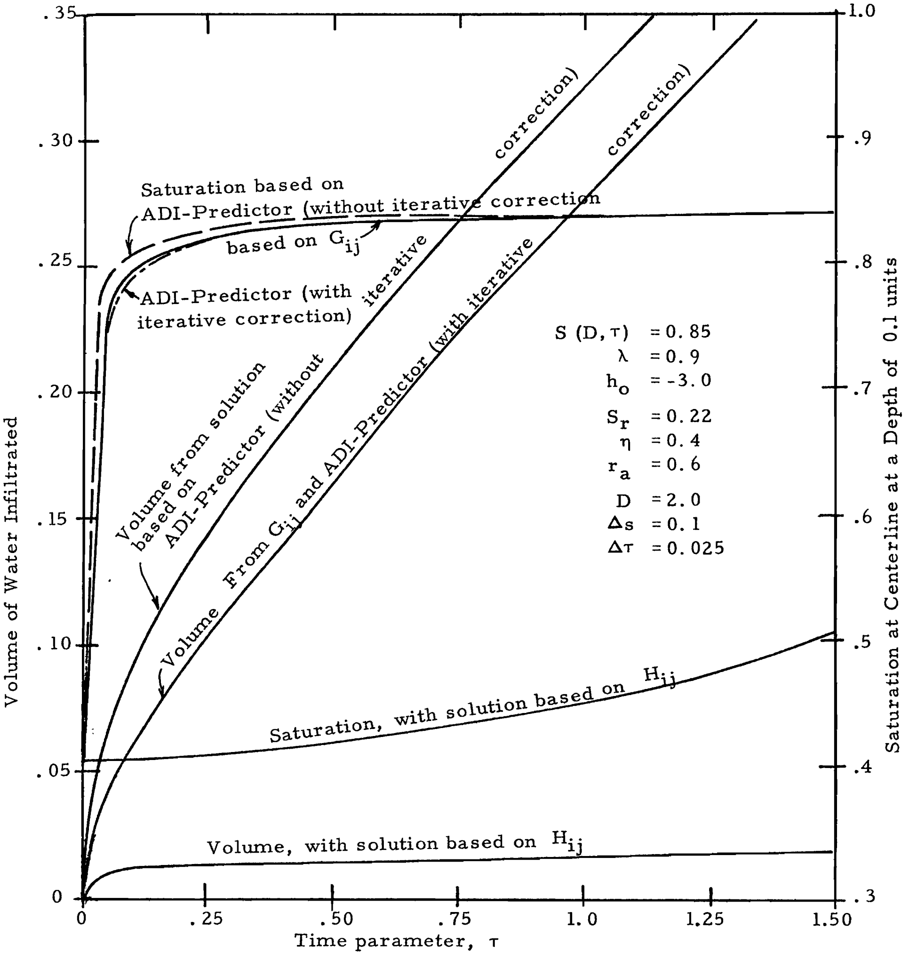


Fig. 9. Intake volume of water and saturation at the centerline at a depth of 0.1 units as given by solutions based on the G_{ij} and H_{ij} finite difference equations and obtained by the ADI-Predictor Method.

whereas the G operator evaluates a_5 (it is denoted as a_4 in Eq. 22) midway between the two time steps before multiplying by $(\xi_{ij}^{k+1} - \xi_{ij}^{k})/\Delta\tau$. Note that the right side of Eq. 49 is actually an approximation of,

$$\frac{\partial(a_5\,\xi)}{\partial\tau} = a_5\,\frac{\partial\xi}{\partial\tau} + \xi\,\frac{\partial a_5}{\partial\tau} \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (50)$$

The vast differences in the solutions based on the H and G operator are caused by the term $\partial a_5/\partial\tau$ in Eq. 50. Particularly for moisture conditions near static equilibrium this term has a large negative magnitude (as can be noted from Fig. 1) which nearly cancels out the positive value of $a_5(\partial\xi/\partial\tau)$, and consequently the solution based on the H operator shows much less change in moisture content than should occur in the infiltration problem.

Since the H operator gives results which bear little resemblance to the solution obtained from the G operator, the ADI method modified by a predictor to evaluate the coefficients at advanced time steps was also studied. The ADI method completes the advance of the solution through a time step by a two step operation. The first step operates along consecutive lines in one direction and the second along lines in the other direction. For axisymmetric infiltration, the finite difference equations which accomplish this for interior grid points are:

$$- b_2\xi_{i-1j}^{k+\frac{1}{2}} + (2 + a_4^{k+\frac{1}{4}})\xi_{ij}^{k+\frac{1}{2}} - b_1\xi_{i+1j}^{k+\frac{1}{2}} = (1 + a_3^k)\xi_{ij-1}^k + (a_4^{k+\frac{1}{4}} - 2)\xi_{ij}^k$$

$$+ (1 - a_3^k)\xi_{ij+1}^k \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (51)$$

for the first portion of the time step, and

$$- (1 + a_3^{k+1})\xi_{ij-1}^{k+1} + (2 + a_4^{k+\frac{3}{4}})\xi_{ij}^{k+1} - (1 - a_3^{k+1})\xi_{ij-1}^{k+1} = b_2\xi_{i-1j}^{k+\frac{1}{2}} + (a_4^{k+\frac{3}{4}} - 2)\xi_{ij}^{k+\frac{1}{2}}$$

$$+ b_1\,\xi_{i+1j}^{k+\frac{1}{2}} \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (52)$$

for the second portion of the time step. The top surface and impervious layer finite difference equations are:

$$- b_2\xi_{i-1,1}^{k+\frac{1}{2}} + (2 + a_4^{k+\frac{1}{4}})\xi_{i1}^{k+\frac{1}{2}} - b_1\xi_{i+1,1}^{k+\frac{1}{2}} = 2\,\xi_{i2}^k + 2\Delta s\,(1 + a_3^k)\,(K_r^k - v_1)$$

$$+ (a_4^{k+\frac{1}{4}} - 2)\xi_{i2}^k \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (53)$$

and

$$- b_2\xi_{i-1N_z}^{k+\frac{1}{2}} + (2 + a_4^{k+\frac{1}{4}})\xi_{iN_z}^{k+\frac{1}{2}} - b_1\,\xi_{i+1N_z}^{k+\frac{1}{2}} = 2\xi_{iN_z-1}^k + (a_4^{k+\frac{1}{4}} - 2)\xi_{iN_z}^k$$

$$- 2\Delta s\,(1 - a_3^k)\,K_r^k \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (54)$$

for the first portion of the time step, and

$$(2+a_4^{k+\frac{3}{4}})\xi_{i1}^{k+1} - 2\xi_{i2}^{k+1} = 2\Delta s\ (1+a_3^{k+1})(K_r^{k+1} - v_1) + b_2\xi_{i-1,1}^{k+\frac{1}{2}} + (a_4^{k+\frac{3}{4}} - 2)\xi_{i1}^{k+\frac{1}{2}}$$

$$+ b_1\xi_{i+1,1}^{k+\frac{1}{2}} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (55)$$

and

$$-2\xi_{i\,N_z-1}^{k+1} + (2+a_4^{k+\frac{3}{4}})\xi_{i\,N_z}^{k+1} = -2\Delta s\ (1-a_3^{k+1})K_r^{k+1} + b_2\xi_{i-1\,N_z}^{k+\frac{1}{2}}$$

$$+ (a_4^{k+\frac{3}{4}} - 2)\xi_{i\,N_z}^{k+\frac{1}{2}} + b_1\xi_{i+1\,N_z}^{k+\frac{1}{2}} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (56)$$

for the second portion of the time step.

The a coefficients in Eqs. 51 through 56 are evaluated after predicting ξ by the technique given in Eq. 48. Thus for example, after predicting $(\xi^{k+\frac{1}{2}})^0$ at the first portion of the time step $a_4^{k+\frac{1}{4}}$ is evaluated by

$$a_4^{k+\frac{1}{2}} = [1 - (.5 + 1.5\lambda)\ \{(\xi^{k+\frac{1}{2}})^0 + \xi^k\}\ ]^{\frac{1}{1+3\lambda}} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (57)$$

and $a_4^{k+\frac{3}{4}}$ is evaluated by using the average of the value $\xi^{k+\frac{1}{2}}$ determined from the first portion of the time step as computed by Eqs. 51, 53 and 55 and the predicted value $(\xi^{k+1})^0$ at the $k+1$ time step. The ADI-predictor method (without the iterative correction) uses only the predicted values with a 0 superscript outside the parenthesis. On Fig. 9 it can be noted that this method gives solutions to the axisymmetric infiltration problem which shows considerably more moisture entering the soil than the solution obtained from the G_{ij} operator. The surface wets up a little more rapidly, but the larger differences in the two solutions occurs within the soil at the wetting front.

In the ADI-predictor method with iterative correction the coefficients are adjusted until the values of $(\xi^{k+\frac{1}{2}})^m$ and $(\xi^{k+1})^m$ are in agreement to within a specified error limit, with the values computed at the first portion and second portions of the time step respectively. This iterative correction equates $(\xi^{k+\frac{1}{2}})^1$ to the $\xi^{k+\frac{1}{2}}$ obtained from solving the system of Eqs. 51, 53 and 55, and then repeats the solution to this system after the improved a's are computed. This process is repeated for each line, first for the first portion of the time step until $(\xi^{k+\frac{1}{2}})^m$ is within specified error tolerance of $\xi^{k+\frac{1}{2}}$, and then the process is repeated for the second portion of the time step. For the limited number of problems solved to date, this process has converged.

On Fig. 9 the volume of water infiltrated into the soil as computed by the ADI-predictor method with iterative correction agrees to approximate four digits with that computed using the Crank-Nicolson method and the G_{ij} finite difference operator. The surface saturation by the ADI-predictor with iterative corrections is slightly less for the first few time steps, however.

Fig. 10 shows a comparison of the saturation at two different points by different methods of solution to a problem in which the infiltration rate was specified as constant. The ADI-predictor with iterative corrections and the Crank-Nicolson method gives essentially identical saturations at all time steps at the surface centerline for this problem. At the other surface point at 0.1 units beyond the circle of application, the ADI-predictor with iterative correction shows slightly greater saturations for the beginning time steps, but for practical purposes the difference between the two solutions is insignificant. The ADI-predictor method without iterative corrections yield a solution that is not in close agreement with the other two.

Table 1 illustrates how the values $(\xi^{k+\frac{1}{2}})^m$ and $(\xi^{k+1})^m$ vary over a few iterations for the first time step in obtaining the solution given in Fig. 10 at three grid points on and close to the surface and consequently essentially at the wetting front. The change in these values with m explains why the iterative corrections are needed. At grid points not within the wetting front, the predictor with $m = 0$ is much closer than at the grid points in Table 1, but it is clear that without a correction the solution may be considerably in error.

The solution results shown in Fig. 10, which are denoted by "ADI method" only, were obtained by evaluating the coefficients which cause the nonlinearities in the differential equation from known values at the current and first portion of the time step as described in Jeppson (1970a). (The one-dimensional solution, based on similar assumptions, is described in Jeppson (1970b).) As shown on

Table 1. Variation of $(\xi^{k+\frac{1}{2}})^m$ and $(\xi^{k+1})^m$ at three grid points in solving the problem of Fig. 10 by the ADI-predictor method with iterative corrections for a few iterations of the first time step.

iteration, m	0	1	2	3	4	5	6	7	8
$(\xi_{2,1}^{k+\frac{1}{2}})^m$	.0220	.0942	.1253	.1406	.1491	.1541	.1572	.1593	.1607
$(\xi_{2,2}^{k+1})^m$	.1536	.1752	.1793	.1805	.1810	.1812	.1813	.1814	.1814
$(\xi_{7,2}^{k+1})^m$	.1627	.1778	.1804	.1811	.1814	.1815	.1815		

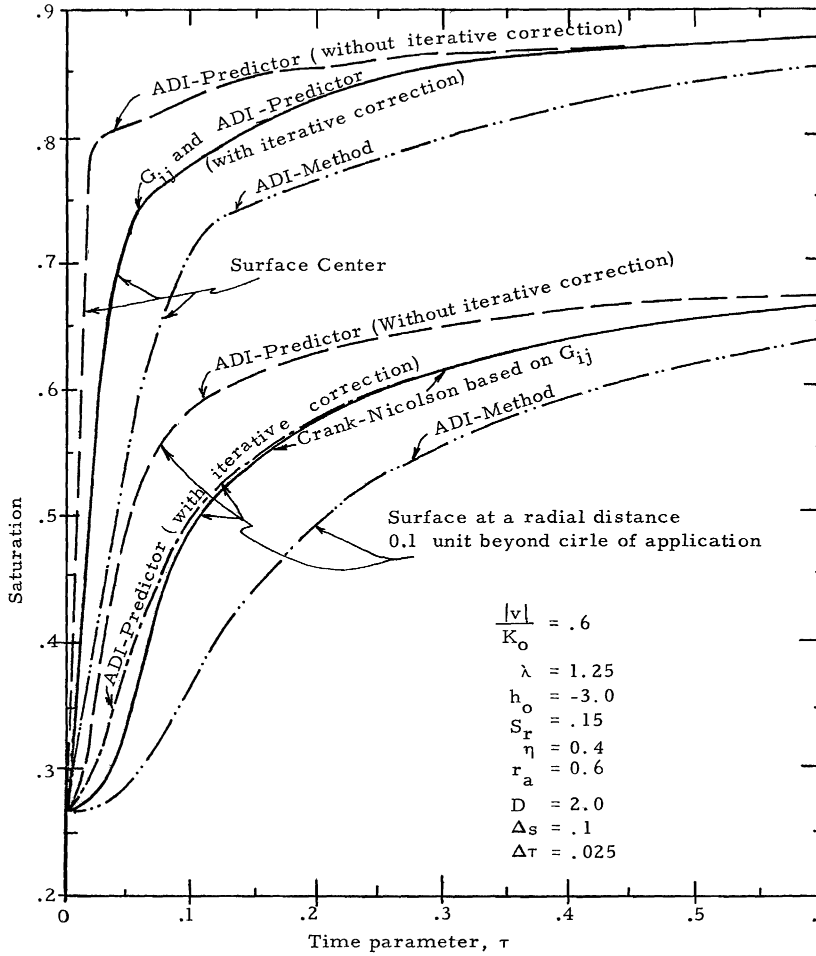


Fig. 10. Saturations on the surface at the centerline and 0.1 units from circle of application obtained from solutions using: (1) The Crank-Nicolson method and the G_{ij} finite difference operators, (2) The ADI-Predictor with iterative correction of coefficients, (3) The ADI-Predictor without iterative correction of coefficients, and (4) The ADI method with coefficients evaluated from knowns at the current and half-time steps.

Fig. 10 considerable discrepancy exists between this solution and that obtained from the Crank-Nicolson method. Note, however, that the differences between the solution results tends to diminish with time, and if the abscissa of Fig 10 were extended less difference would exist between the given saturations.

In contrast with the study described earlier in reducing the time step size $\Delta\tau$ and the space increments Δs in solutions by the Crank-Nicolson method, neither the ADI-predictor without interative correction nor the ADI method reproduces solutions for varying time steps or space increments. Rather these solutions tend more toward those given by the Crank-Nicolson method and the ADI-predictor method with iterative corrections when the increments are made smaller.

The Crank-Nicolson method requires slightly more computer execution time than the ADI method with iterative correction for comparable problems solved. This comparison is based on solutions that required the sum of changes between consecutive Newton-Raphson iterations be less than 3×10^{-8} before terminating the solution and terminating the corrective iteration in the ADI-predictor method when the largest change between $(\xi)^{m}$ and $(\xi)^{m-1}$ was less than 0.0001. No particular numerical difficulties occurred in obtaining the solutions to the G_{ij} finite difference operators, but in using the ADI-predictor method it was noted that the values of ξ at grid point at radial distances beyond the wetting front, particularly near the surface tended to become slightly larger than the initial static equilibrium values. Logic was therefore added into the computer solution to set any values of ξ which were computed larger than the static equilibrium value equal to the static equilibrium value. Since no such constraints were programmed into solving the Crank-Nicolson method, using the G_{ij} operator, the writer favors this method slightly but also because there is less question regarding the order of approximation from use of the Crank-Nicolson method.

SUMMARY AND CONCLUSIONS

Transient, unsaturated vertical moisture movement and two-dimensional axisymmetric moisture movement in soils due to infiltration on the surface are solved using adaptations of common finite difference methods which were developed and whose performance has been studied over the years for solving linear initial-boundary-value problems. Three different differencing schemes are used in applying the Crank-Nicolson method. One of these schemes uses a time difference which is valid only for constant coefficients, and it produces a solution which is grossly in error. The other two schemes use essentially similar differences, but in one case the coefficient of the time derivative is divided into the equation before differencing. While these latter two schemes produce practically identical solutions, the scheme which retains the coefficient with the time derivative supplies a system of nonlinear algebraic equations which is much easier to solve by the Newton-Raphson iterative method than those supplied by the other scheme. The functional variation of the finite difference equations obtained after multiplying the space derivatives by the reciprocal of the time derivative coefficient is so rapid in the neighborhood of the solution vector, particularly for conditions near static equilibrium, that the Newton-Raphson iteration is inadequate. Only upon using a Fibonacci search-iterative scheme to provide a very close initial guess (or some other scheme) will the Newton-Raphson method converge to the solution of this nonlinear system of finite difference equations.

Three variations of the ADI method are also used in solving the two-dimensional problems. Only when the coefficients in the finite difference equations produced by the ADI-method are iteratively corrected are the solutions from this method in close agreement with those produced by the Crank-Nicolson method. The discrepancy between the solutions is confined primarily to the region around the wetting front where the variables of the problem change rapidly with both space and time. These differences, however, cause many other features of the solutions to be different. Use of a predictor to evaluate the coefficients at advanced time steps without correcting the predicted values produces a solution only slightly better than evaluating the coefficients entirely from known values at current time steps.

Since the solutions based on the Crank-Nicolson methods and the ADI-predictor methods with iterative corrections produce essentially identical solutions and these solutions are reproducible using smaller time and space increments, it appears that the finite difference solutions do converge to the true solution of the initial-boundary value problem governed by the strongly nonlinear equation of flow.

The fact that sizable differences in solutions (and even grossly erroneous solutions) are due to the methods of approximation used in the other methods, emphasizes the need for meticulous concern about how the partial differential equation is differenced. The equation of flow is strongly nonlinear when relationships for saturation, hydraulic conductivity and capillary pressure are used which describe the hydraulic properties of real soils. The nonlinearities of the equations cause not only the differences in solutions from different methods, but require that special techniques be used to solve the finite difference equations. The ease with which the finite difference equations, which are obtained by slight variations in the mathematical procedure followed in developing them, can be solved is vastly different. The G_{ij} finite difference operators used in this study are much easier to solve than the F_{ij} finite difference operators. Since only those methods which implicitly evaluate the coefficients at advanced time steps produce solutions which are in agreement, it is doubtful whether an explicit method for solving parabolic partial differential equations would be adequate.

A theory of methods exclusively developed and analyzed for solving nonlinear equations is needed. In the absence of this theory, it appears that methods developed for solving problems associated with linear partial differential equations, can produce accepted solutions, but only if these methods are appropriately modified to cope with the effects of the nonlinearities.

REFERENCES

1. Ames, W. F. 1965. Nonlinear Partial Differential Equations in Engineering. Academic Press.

2. Brooks, R. H. and A. T. Corey. 1966. "Properties of Porous Media Affecting Fluid Flow." Journal of the Irrigation and Drainage Division, ASCE, Vol. 92, No. IR2, Proc. Paper 4855, June. pp. 61-87.

3. Brooks, R. H., B. Ng, G. L. Corey, and A. T. Corey. 1971. "Drainage of Soil Profiles." Journal of the Irrigation and Drainage Division, ASCE, Vol. 97, No. IR3, Proc. Paper 8389, Sept. pp. 455-467.

4. Brutsaert, W. F. 1971. "A Functional Iteration Technique for Solving the Richards Equations Applied to Two-Dimensional Infiltration Problems." Water Resources Research, Vol. 7, No. 6, Dec. pp. 1583-1596.

5. Bryan, C. A. 1964. "On the Convergence of the Method of Nonlinear Simultaneous Displacements." Rendiconti del Circolo Matematico di Palermo, Serie II, Tomo XIII.

6. Corey, G. L. and A. T. Corey. 1967. "Similitude of Drainage of Soils." Journal of the Irrigation and Drainage Division, ASCE, Vol. 93, No. IR3, Proc. Paper 5462, Sept. pp. 3-23.

7. Douglas, Jim, Jr. 1961. "A Survey of Numerical Methods of Parabolic Differential Equations." Advances in Computers, Vol. 2, Academic Press, New York.

8. Freeze, R. A. 1971. "Three-Dimensional, Transient, Saturated-Unsaturated Flow in a Groundwater Basin." Water Resources Research, Vol. 7, No. 2, April. pp. 347-366.

9. Hanks, R. J. and S. A. Bowers. 1962. "Numerical Solution of the Moisture Flow Equation for Infiltration into Layered Soils." Soil Science Society of America, Proceedings, Vol. 26, No. 6, Nov-Dec. pp. 530-534.

10. Jeppson, R. W. 1970a. "Formulation and Solution of Transient Flow of Water from an Infiltrometer Using the Kirchhoff Transformation." PRWG-59c-3, Utah Water Research Laboratory, Utah State University, Logan, Utah, July. (Also see PRWG-59c-2.)

11. Jeppson, R. W. 1970b. "Solution to Transient Vertical Moisture Movement Based upon Saturation-Capillary Pressure Data and a Modified Burdine Theory." PRWG-59c-5, Utah Water Research Laboratory, Utah State University, Logan, Utah, December.

12. King, L. G. 1965. "Description of Soil Characteristics for Partially Saturated Flow." Soil Science Society of America, Proceedings, Vol. 29, No. 4, July-August. pp. 359-362.

13. Lieberstein, H. M. 1959. "Overrelaxation for Non-Linear Elliptic Partial Differential Problems." Tech. Sum. Rept. No. MRC-TSR-80, Univ. of Wisconsin Math. Res. Center, Madison, Wisconsin.

14. Raats, P. A. C. and W. R. Gardner. 1971. "Comparison of Empirical Relationship between Pressure Head and Hydraulic Conductivity and Some Observations on Radially Symmetric Flow." Water Resources Research, Vol. 7, No. 4, August. pp. 921-928.

15. Remson, I. R., G. M. Hornberger and F. J. Molz. 1971. Numerical Methods in Subsurface Hydrology. Wiley-Interscience, New York.

16. Rubin, Jacob. 1968. "Theoretical Analysis of Two-Dimensional, Transient Flow of Water in Unsaturated and Partly Unsaturated Soils." Soil Science Society of America, Proceedings, Vol. 32, No. 5, pp. 607-615.

17. Saaty, T. L. and J. Bram. 1964. Non-Linear Mathematics. McGraw-Hill Book Co., Inc.

18. Smith, R. E. and D. A. Woolhiser. 1971. "Mathematical Simulation of Infiltration Watersheds." Hydrology Paper 47, Colorado State Univ., Fort Collins. (Also see "Overland Flow on an Infiltration Surface." Water Resources Research, Vol. 7, No. 4, Aug. 1971, pp. 899-913.)

19. Thurman, Donald H. 1963. "Algorithm 195 BANDSOLUE." Association for Computing Machinery. Vol. 6, No. 8.

20. Verna, R. D. and W. F. Brutsaert. 1970. "Unconfined Aquifer Seepage by Capillary Flow Theory." Journal of the Hydraulics Division, ASCE, Vol. 96, No. HY6, Proc. Paper 7363, June. pp. 1331-1344.

21. Wilde, D. J. and C. S. Beighter. 1967. Foundations of Optimization. Prentice-Hall, Inc.

APPENDIX A

LISTING OF FORTRAN PROGRAM WHICH UTILIZES THE G FINITE DIFFERENCE OPERATOR TO SOLVE AXISYMMETRIC INFILTRATION PROBLEMS

```
@FOR,IS  TRANSC,TRANSC
        INTEGER NTAPE(32)
      COMMON H(32,32),B(32,32),DM(32),D(32),DP(32),F(32),S1(32),R1(32),
     $R2(32),Q,W,VK,HEIGT,DEPTH,SR,SR1,EXP1,PB,POR,AMBDA,DELT,DELS,HI,
     $HIT,QF,QF1,DELSCB,DELSC2,DCUB,ERR,ERR1,DCUB5,TIME,COFA1,DELS2,DES2
     $,DELH,AMB31,AMB21,RAMB31,EXP2,EXP3,AMB32,COFA,SSUR,MY1,NX,NY,NX1,
     $NY1,NX2,NY2,N2X1,N2X,N2XP,NHSR,MY,NXX1,MX,NSAT,NSSUR,NB1,NB2,MB2,
     $MB3,MAX,MAXH,HD(32,32),WATE1,AREAC,AMB3H,AMB2H,AMB22,EXP4,AC2,TM1
        NSKIP=0
      INCRT=0
   1 0 READ(5,100,END=99) N2X,MX,MY,NT,HI,DEPTH,DELT,Q,PB,SSUR
      IF(N2X .GT. 50) GO TO 99
      TM1=0.
      WATE1=0.
      NSSUR=0
      IF(SSUR .GT. .001) NSSUR=1
      NXX1=N2X+1
      MY1=MY-1
      HEIGT=DEPTH/PB
      HIT=HI/PB
      IF(NSKIP.EQ. 1) WRITE(6,210)
  21 0 FORMAT(1H1)
      IF(NSSUR .GT. 0) GO TO 8
      WRITE(6,102) N2X,MX,MY,NT,HI,HIT,DEPTH,HEIGT,DELT,Q,PB
  10 2 FORMAT(' N2X=',I3,' NX=',I3,' NY=',I3,' NT=',I5,' HI=',F8.3,' HIT=
     $',F8.3,' DEPTH=',F7.2,' HEIGT=',F8.3,' DELT=',F8.4,' Q=',F10.6,'
     $ PB=',F8.3)
      GO TO 9
  8   WRITE(6,204) N2X,MX,MY,NT,HI,HIT,DEPTH,HEIGT,DELT,SSUR,PB
  20 4 FORMAT(' N2X=',I3,' NX=',I3,' NY=',I3,' NT=',I5,' HI=',F8.3,' HIT=
     $',F8.3,' DEPTH=',F7.2,' HEIGT=',F8.3,' DELT=',F8.4,' S(1)=',F8.4,'
     $ PB=',F8.3)
 10 0 FORMAT(4I5,6F10.5)
    9 READ(5,101) SR,AMBDA,POR,DTINC,EXPAND,ERR,VK
      ERR1=.1*ERR
 10 1 FORMAT(8F10.5)
      READ(5,100) NRIT1,NRIT2,INCB,NHSTAR,STREAD
      READ(5,176) NWRIT,NWR2,NSAT,NTABLE ,MAX
      MAXH=MAX/2
 17 6 FORMAT(16I5)
      WRITE(6,391)  NRIT1,NRIT2,INCB,NSSUR,MAX,NWRIT,NWR2
 39 1 FORMAT(' NRIT1=',I2,' NRIT2=',I2,' INCB=',I2,' NSSUR=',I2,' MAX=',
     $I2,I5,' TIME PLANE  SOLUTIONS THAT WILL BE STORED ON TAPE',I2,/,
     $'  TIME STEPS STORED ARE')
      N2XP=N2X+1
       NMOST=1
      IF(NSSUR .GT. 0) GO TO 34
      NB1=1
      NB2=2
      MB2=2
      MB3=3
      GO TO 32
  3 4 NB1=2
      NB2=3
      MB2=N2XP
```

```
      MB3=MB2+1
   32 WRITE(6,103) SR,AMBDA,POR,ERR
  103 FORMAT(' SR=',F8.3,' LAMBDA=',F7.2,' POROSITY=',F8.3,' ERR=',E9.3)
      IF(NWRIT .LT. 1) GO TO 33
      READ(5,176)(NTAPE(I),I=1,NWRIT)
      WRITE(6,177) (NTAPE(I),I=1,NWRIT)
  177 FORMAT(1H ,32I4)
   33 NY=6
      IF(HEIGT-HIT .GT. .99) GO TO 45
      WRITE(6,179)
  179 FORMAT('0PROBLEM SPECIFICATIONS OUTSIDE RANGE OF VALIDITY OF BROOK
     $S-COREY EQUATIONS')
      GO TO 10
   45 IF(HIT .LT. -.99) GO TO 46
      WRITE(6,180)
  180 FORMAT('0BEFORE WATER PENETRATES TO BOTTOM PROBLEM SPECIFICATIONS
     $WILL BE OUTSIDE THE RANGE OF VALIDITY OF THE BROOKS-COREY EQS.')
   46 DO 47 I=1,MX
      DO 47 J=1,MY
   47 HD(I,J)=0.
      ITAPE=1
      NX=N2X+6
      QF1=0.
      DELS=HEIGT/FLOAT(MY-1)
      SR1=1.0-SR
      IF(STREAD .LT. 1.0) GO TO 39
      READ(5,100) NREC,NFIL,NREC1,NFI1,QF1
      IF(NREC .GT. 0 .OR. NFIL .GT. 0) CALL SKPFLS(NWR2,NFIL,NREC)
      DO 40 I=2,MX
      W=.5/FLOAT(I-1)
      R1(I)=1.+W
   40 R2(I)=1.-W
      IF(NREC1 .GT. 0) CALL SKPFLS(NWR2,0,NREC1)
      CALL INOUT(1,NWR2,H(1,1),1024)
      IF(NFI1 .GT. 0) CALL SKPFLS(NWR2,NFI1,0)
      READ(5,100) NX,NY
      DO 44 J=1,MY
      W=1./(HEIGT-DELS*FLOAT(J-1)-HIT)
   44 S1(J)=SR+SR1*W**AMBDA
   39 N2X1=N2X-1
      NX1=NX-1
      NY1=NY-1
      FY=NY1
      NX2=NX-2
      NY2=NY-2
      W=DELS*FLOAT(N2X1)
      AREAC=3.141659*W*W
      IF(VK .LT. .001) GO TO 11
      Q=VK*AREAC
      IF(NSSUR .EQ. 0) WRITE(6,109) VK,W,Q,DELS,AREAC
  109 FORMAT(' FLUX PER UNIT AREA HAS BEEN SPECIFIED EQUAL TO',F8.4,' RA
     $DIUS=',F8.4,' Q/KO=',F8.4,' DELS=',F8.3,' AREA=',F8.3)
      GO TO 12
   11 VK=Q/AREAC
      IF( NSSUR .GT. 0) WRITE(6,108) W,VK,DELS,AREAC
  108 FORMAT(' RADIUS OVER WHICH INFILTRATION OCCURS' , F10.5,' INFILTRA
     $TION FLUX =',F9.4,' DELS =',F8.3,' AREA=',F8.3)
   12 DELSCB=6.2831853*POR*DELS**3
      DELSC2=.5*DELSCB
      DCUB=19.634954*POR*DELS**3
      DCUB5=.5*DCUB
      DES2=2.*DELS
```

```
      DELH=.5*DELS
      AMB31=1.+3.*AMBDA
      AMB21=1.+2.*AMBDA
      RAMB31=1./AMB31
      EXP1=AMB21*RAMB31
      AMB32=1.+AMB31
      AMB2H=.5*AMB21
      AMB22=AMB21+1.
      EXP4=RAMB31*AMB22
      AC2=DELH*AMB32
      AMB3H=.5*AMB31
      COFA1=DELS*AMB32
      COFA=DELH*AMB32
      EXP2=AMB32*RAMB31
      EXP3=AMBDA*RAMB31
      DELSS=2.*DELS*DELS/DELT
      QF=POR*SR1*AMBDA*PB
    7 DET=DELT
      DELT=DELT/FLOAT(INCB)
      DELS2=DELSS*FLOAT(INCB)
      TIME=0.0
      IF(STREAD .LT. 1.) CALL INITIA
      DO 1 I=1,INCB
      NHSR=0
      NRR=MOD(I,NRIT1)
      CALL TIMSTH
      TIME=TIME+DELT
      IF(NRR .GT. 0) GO TO 1
      IF(NHSTAR .LT. 1) WRITE(6,104) TIME
  104 FORMAT('0VALUES OF THE XI FOR TIME=',F12.5)
      CALL RITOUT(NHSTAR,I)
    1 CONTINUE
      DELT=DET
      DELS2=DELSS
      KTIM=1
      DO 2 I=2,NT
      IF(INCRT .EQ. 0) GO TO 71
      INCRT=0
      DELT=1.5*DELT
      WRITE(6,174) I,DELT
  174 FORMAT(' TIME STEP=',I5,' DELT=',F13.6)
      DELS2=DELS2/1.5
   71 NHSR=0
      NRR=MOD(I,NRIT2)
      HTEM=H(3,2)
      CALL TIMSTH
      TIME=TIME+DELT
      IF(NRR .GT. 0) GO TO 3
      IF(NHSTAR .LT. 1) WRITE(6,104) TIME
      CALL RITOUT(NHSTAR,I)
    3 NY2=NY-2
      IF(H(2,NY2) .GT. H(MX,NY2)-EXPAND .OR. NY .EQ. MY) GO TO 4
      NY=NY+1
      GO TO 3
    4 NY1=NY-1
      NY2=NY-2
      FY=NY1
    5 NX4=NX-4
      IF(H(NX4,1) .GT. H(MX,1)-EXPAND .OR. NX .EQ. MX) GO TO 6
      NX=NX+1
      GO TO 5
    6 NX1=NX-1
```

```
      NX2=NX-2
    2 IF(NWRIT .LT. 1) GO TO 2
      IF(I .NE. NTAPE(ITAP)) GO TO 2
      IF(ITAP .GT. NWRIT) GO TO 2
      WRITE(6,252) ITAP
  252 FORMAT(' FILE NO.',I5, ' HAS BEEN WRITTEN ON TAPE')
      ITAP=ITAP+1
      CALL INOUT(0,NWR2,H(1,1),1024)
    2 CONTINUE
      NSKIP=1
      IF(NWRIT .GT. 0) CALL ENFILE(NWR2)
      GO TO 10
   99 IF(NWRIT .GT. 0 .AND. N2X .EQ. 99) CALL UNLOAD(NWR2)
      STOP
      END
@FOR,IS OUTPTC,OUTPTC
      SUBROUTINE RITOUT(NM,ITIME)
      COMMON H(32,32),B(32,32),DM(32),D(32),DP(32),F(32),S1(32),R1(32),
     $R2(32),Q,W,VK,HEIGT,DFPTH,SR,SR1,EXP1,PB,POR,AMBDA,DELT,DFLS,HI,
     $HIT,QF,QF1,DELSCB,DELSC2,DCUB,ERR,ERR1,DCUBS,TTME,COFA1,DELS2,DES2
     $,DFLH,AMB31,AMB21,RAMB31,EXP2,EXP3,AMB32,COFA,SSUR,MY1,NX,NY,NX1,
     $NY1,NX2,NY2,N2X1,N2X,N2XP,NHSR,MY,NXX1,MX,NSAT,NSSUR,NB1,NB2,MB2,
     $MB3,MAX,MAXH,HD(32,32),WATE1,AREAC,AMB3H,AMB2H,AMB22,EXP4,AC2,TM1
      IF(NM .GT. 0) GO TO 35
      NM1=1
      NM2=16
    1 IF(NM2 .GT. NX) NM2=NX
      WRITE(6,101) (I,I=NM1,NM2)
  101 FORMAT(3H    ,I5,15I8)
      DO 2 J=1,NY
    2 WRITE(6,100) J,(H(I,J),I=NM1,NM2)
  100 FORMAT(1H ,I2,16F8.4)
      IF(NM2 .EQ. NX) GO TO 3
      NM1=NM1+16
      NM2=NM2+16
      GO TO 1
    3 IF(NM .LT. 0) RETURN
    5 WRITE(6,104) ITIME,TIME
  104 FORMAT('0 VALUES OF SATURATION FOR TIME STEP',I5,' TAU =',F9.4)
      HIM=HI+.0003
      RPT=(1.-AMB31*H(2,1))**RAMB31
      D(2)=SR+SR1*RPT**AMBDA
      D(1)=D(2)
      B(2,1)=HEIGT-1./RPT
      WATCOT=DCUBS*(D(2)-S1(1))
      I=2
      J=1
    1 I=I+1
      RPT=(1.-AMB31*H(I,1))**RAMB31
      D(I)=SR+SR1*RPT**AMBDA
      WATCOT=WATCOT+DELSC2*(D(I)-S1(J))*FLOAT(I-1)
      B(I,J)=HEIGT-1./RPT
      IF(B(I,J) .GT. HIM .AND. I .LT. NX) GO TO 11
      IMAX=I
      B(1,J)=I
      WRITE(6,103) J,(D(II),II=1,I)
   12 J=J+1
      XI=HEIGT-DFLS*FLOAT(J-1)
      RPT=(1.-AMB31*H(2,J))**RAMB31
      D(2)=SR+SR1*RPT**AMBDA
      D(1)=D(2)
      WATCOT=WATCOT+DCUB*(D(2)-S1(J))
```

```
      DPE=XI-1./RPT
      B(2,J)=DPE
      I=2
   13 I=I+1
      RPT=(1.-AMB31*H(I,J))**RAMB31
      D(I)=SR+SR1*RPT**AMBDA
      B(I,J)=XI-1./RPT
      WATCOT=WATCOT+DELSCB*(D(I)-S1(J))*FLOAT(I-1)
      IF(B(I,J) .GT. HIM .AND. I .LT. NX1) GO TO 13
      IF(I .GT. IMAX) IMAX=I
      B(1,J)=I
      WRITE(6,103) J,(D(II),II=1,I)
  103 FORMAT(1H ,I2,10(16F8.4,/,1H ))
      IF(DPE .GT. HIM .AND. J .LT. NY1) GO TO 12
      IF(J .LT. MY1) GO TO 29
      RPT=(1.-AMB31*H(I,MY))**RAMB31
      D(2)=SR+SR1*RPT**AMBDA
      D(1)=D(2)
      B(2,J)=-1./RPT
      J=MY
      WATCOT=WATCOT+DCUB5*(D(2)-S1(MY))
      I=2
   20 I=I+1
      RPT=(1.-AMB31*H(I,MY))**RAMB31
      D(I)=SR+SR1*RPT**AMBDA
      B(I,J)=-1./RPT
      WATCOT=WATCOT+DELSC2*(D(I)-S1(MY))*FLOAT(I-1)
      IF(B(I,J) .GT. HIM .AND. I .LT. NX1) GO TO 20
      B(1,J)=I
      WRITE(6,103) MY,(D(II),II=1,I)
   29 XI=PB*WATCOT
      RATE=WATCOT/((TIME+QF1)*QF)
      RPT=PB*RATE
      D(1)=(WATCOT-WATE1)/(QF*(TIMF-TM1))
      D(2)=D(1)/AREAC
      D(3)=D(2)*PB
      WRITE(6,102) WATCOT,XI,(D(I),I=1,3),RATE,RPT,ITTIME,TIME
  102 FORMAT(' VOL OF WATER ABSORBED =',2F10.6,' Q=',F10.5,' RATE=',2F9.
     $4,' ACCUM. Q=',2F9.4,/,'D VALUES FOR HYDRAULIC HEAD FOR TIME STEP'
     $,I5,' TAU =',F9.4)
      TM1=TIME
      WATE1=WATCOT
      DO 24 JJ=1,J
      II=B(1,JJ)+.01
   24 WRITE(6,106) JJ,B(2,JJ),(B(I,JJ),I=2,II)
  106 FORMAT(1H ,I2,16F8.3,3(/,3X,16F8.3))
      RETURN
      END
 ø .IS  STARTC,STARTC
      SUBROUTINE INITIA
      COMMON H(32,32),B(32,32),DM(32),D(32),DP(32),F(32),S1(32),R1(32),
     $R2(32),Q,W,VK,HEIGT,DEPTH,SR,SR1,EXP1,PB,POR,AMBDA,DELT,DELS,HI,
     $HIT,QF,QF1,DELSCB,DELSC2,DCUB,ERR,ERR1,DCUB5,TIMF,COFA1,DELS2,DES2
     $,DELH,AMB31,AMB21,RAMB31,EXP2,EXP3,AMB32,COFA,SSUR,MY1,NX,NY,NX1,
     $NY1,NX2,NY2,N2X1,N2X,N2XP,NHSR,MY,NXX1,MX,NSAT,NSSUR,NB1,NB2,MB2,
     $MB3,MAX,MAXH
      NNCT=0
      DO 2 J=1,MY
      RPT=1./(HEIGT-DELS*FLOAT(J-1)-HIT)
      S1(J)=SR+SR1*RPT**AMBDA
      XI=RAMB31*(1.-RPT**AMB31)
      DO 2 I=1,MX
```

```
    2 H(I,J)=XI
      XI=H(2,1)
      A1=COFA*(1.-AMB31*XI)**RAMB31
      B(32,1)=2.*(2.*(H(2,2)-XI)+DES2*(1.+A1)*(1.-AMB31*XI)**EXP2)
      DO 4 J=2,MY1
      A1=COFA*(1.-AMB31*H(2,J))**RAMB31
    4 B(32,J)=2.*((1.+A1)*H(2,J-1)+(1.-A1)*H(2,J+1)-2.*H(2,J))
      WRITE(6,203) (B(32,J),J=1,MY1)
  203 FORMAT(1H ,11E11.5,E10.5)
      WRITE(6,202) (H(2,J),J=1,MY)
  202 FORMAT(' INITIAL DISTRIBUTION OF XI THRU PROFILE',5(/1H ,13F10.7))
      WRITE(6,201) (S1(J),J=1,MY)
  201 FORMAT(' INITIAL SATURATION THRU PROFILE',5(/,1H ,13F10.4))
      RPT=N2X
      DO 1 I=2,MX
      XI=0.5/FLOAT(I-1)
      R1(I)=XI+1.
    1 R2(I)=1.-XI
      R1(1)=11.
      R2(1)=9.
      IF(NSSUR .EQ. 0) RETURN
      ARG=AMB31/AMBDA
      XI=RAMB31*(1.-((SSUR-SR)/SR1)**ARG)
      DO 25 I=1,N2X
   25 H(I,1)=XI
      RETURN
      END
@FOR,IS  STEPTC,STEPTC
      SUBROUTINE TIMSTH
      COMMON H(32,32),B(32,32),DM(32),D(32),DP(32),F(32),S1(32),R1(32),
     $R2(32),Q,W,VK,HEIGT,DEPTH,SR,SR1,EXP1,PB,POR,AMBDA,DELT,DELSI,HI,
     $HIT,QF,QF1,DELSCB,DELSC2,DCUB,ERR,ERR1,DCUB5,TIMF,COFA1,DELS2,DES2
     $,DELH,AMB31,AMB21,RAMB31,EXP2,EXP3,AMB32,COFA,SSUR,MY1,NX,NY,NX1,
     $NY1,NX2,NY2,N2X1,N2X,N2XP,NHSR,MY,NXX1,MX,NSAT,NSSUR,NB1,NB2,MB2,
     $MB3,MAX,MAXH,HD(32,32),WATE1,AREAC,AMB3H,AMB2H,AMB22,EXP4,AC2
    1 VVK=VK
      B0=B(32,1)
      DO 2 I=MB2,NX1
      IF(I .EQ. N2XP) VVK=0.
      H0=H(I,1)
      ARG=1.-AMB31*H0
      A1=COFA*ARG**RAMB31
      RK=ARG**EXP2
    2 B(I,1)=R1(I)*H(I+1,1)+R2(I)*H(I-1,1)+2.*H(I,2)-4.*H0+DES2*(1.
     $+A1)*(RK-VVK)-B0
C    $+A1)*(RK-VVK)
      DO 3 J=2,NY1
      B0=B(32,J)
      JM=J-1
      JP=J+1
      DO 3 I=2,NX1
      H0=H(I,J)
      ARG=1.-AMB31*H0
      A1=COFA*ARG**RAMB31
    3 B(I,J)=R1(I)*H(I+1,J)+R2(I)*H(I-1,J)+(1.+A1)*H(I,JM)+(1.-A1)*H(I,
     $JP)-4.*H0-B0
C    $JP)-4.*H0
      NNY=NY1
      IF(NY .LT. MY) GO TO 15
      DO 16 I=2,NX1
      H0=H(I,MY)
      ARG=1.-AMB31*H0
```

```
      A1=COFA*ARG**RAMB31
      RK=ARG**EXP2
    6 B(I,MY)=R1(I)*H(I+1,MY)+R2(I)*H(I-1,MY)+2.*H(I,MY1)-4.*HO-DES2*(1.
     $-A1)*RK
      NNY=MY
    5 NCOUNT=0
      DO 13 J=1,NY1
      DO 13 I=1,NX1
      ARG=H(I,J)
      H(I,J)=ARG-HD(I,J)
   13 HD(I,J)=ARG
    4 SUMT=0.
      DO 5 J=1,NNY
      JP=J+1
      JM=J-1
      NCT=0
      MM2=MB2
      MM3=MB3
    6 IF(J .GT. 1) GO TO 8
      VVK=VK
      DO 7 I=MB2,NX1
      IF(I .EQ. N2XP) VVK=0.
      HO=H(I,J)
      ARG=1.-AMB31*HO
      A1=COFA*ARG**RAMB31
      RK=ARG**EXP2
      ARG1=1.-AMB3H*(HD(I,J)+HO)
      C1=DELS2/ARG1**EXP1
      BO=HO-HD(I,J)
      F(I)=R1(I)*H(I+1,J)+R2(I)*H(I-1,J)+2.*H(I,JP)-4.*HO+DES2*(1.+A1)*
     $(RK-VVK)+B(I,J)-C1*BO
      DM(I)=R2(I)
      D(I)=DES2*((VVK-RK)*A1-(1.+A1)*AMB32*RK)/ARG-C1*AMB2H*BO/ARG1-C1
     $-4.
      DP(I)=R1(I)
      IF(MB2 .EQ. 2) D(2)=D(2)+DM(2)
      GO TO 10
    8 MM2=2
      MM3=3
      IF(J .EQ. MY) GO TO 18
      DO 9 I=2,NX1
      HO=H(I,J)
      H2=H(I,JM)
      H4=H(I,JP)
      ARG=1.-AMB31*HO
      A1=COFA*ARG**RAMB31
      ARG1=1.-AMB3H*(HD(I,J)+HO)
      C1=DELS2/ARG1**EXP1
      BO=HO-HD(I,J)
      F(I)=R1(I)*H(I+1,J)+R2(I)*H(I-1,J)-4.*HO+B(I,J)+(1.+A1)*H2+(1.-A1)
     $*H4-C1*BO
      DM(I)=R2(I)
      D(I)=A1*(H4-H2)/ARG-C1*AMB2H*BO/ARG1-C1-4.
    9 DP(I)=R1(I)
      D(2)=D(2)+DM(2)
      IF(J .LT. MY) GO TO 10
    8 DO 17 I=2,NX1
      HO=H(I,MY)
      ARG=1.-AMB31*HO
      A1=COFA*ARG**RAMB31
      RK=ARG**EXP2
      ARG1=1.-AMB3H*(HD(I,MY)+HO)
```

```
      C1=DELS2/ARG1**EXP1
      BO=HO-HD(I,MY)
      F(I)=R1(I)*H(I+1,MY)+R2(I)*H(I-1,MY)+2.*H(I,MY1)-4.*HO-DES2*(1.-A1
     $ )*RK+B(I,MY)-C1*BO
      DM(I)=R2(I)
      D(I)=DES2*(((1.-A1)*AMB32-A1)*RK )/ARG-C1*AMB2H*BO/ARG1-C1-4.
    7 DP(I)=R1(I)
      D(2)=D(2)+DM(2)
   10 DO 11 I=MM3,NX1
      IM=I-1
      ARG=DM(I)/D(IM)
      F(I)=F(I)-ARG*F(IM)
   11 D(I)=D(I)-ARG*DP(IM)
      I=NX1
      DIF=F(I)/D(I)
      H(I,J)=H(I,J)-DIF
      SUM=ABS(DIF)
   12 I=I-1
      DIF=(F(I)-DP(I)*DIF)/D(I)
      H(I,J)=H(I,J)-DIF
      SUM=SUM+ABS(DIF)
      IF(I .GT. MM2) GO TO 12
      SUMT=SUMT+SUM
      H(1,J)=H(2,J)
      NCT=NCT+1
      IF(SUM .GT. ERR1  .AND. NCT .LT. MAXH) GO TO 6
    5 CONTINUE
      NCOUNT=NCOUNT+1
      IF(SUMT .GT. ERR .AND. NCOUNT .LT. MAX ) GO TO 4
      IF(NCOUNT .EQ. MAXH) WRITE(6,100) NCT,NCOUNT,SUMT
  100 FORMAT(1H ,I3,'  DID NOT CONVERGE IN ALLOWABLE NUMBER OF ITERATION
     $S',I3,'  SUMT=',E15.8)
      DO 14 J=1,NY1
      DO 14 I=1,NX1
    4 HD(I,J)=HD(I,J)-H(I,J)
      RETURN
      END
IXQT
    7   20   21   75   -3.0      2.0       .025      .0      1.0      0.95
  .15      1.5        .4        .361      .00001 .0000003   0.
    1    2    4    1   0.0
   11    8    1    0   20
    2    5   10   15   20   25   30   40   50   60   75
    7   20   21   75   -3.0      2.0       .025      .6      1.0      0.0
  .15      1.5        .4        .361      .00001 .0000003   0.6
    1    2    4    1   0.0
   11    8    1    0   20
    2    5   10   15   20   25   30   40   50   60   75
FIN
@  REMOTE STOP
```

APPENDIX B

DESCRIPTION OF INPUT DATA REQUIRED BY FORTRAN PROGRAM

Card No. 1 (4I5, 6F10.5)

N2X - number of grid points in the radial direction to outer edge of circle r_a over which moisture is applied.

MX - number of grid points in radial direction to outer radius of problem.

MY - number of grid points in axial direction between top surface and bottom of problem. This number must be one greater than the number of space increments through depth of profile.

NT - number of time steps through which computations are to be completed.

HI - value of the static equilibrium initial hydraulic head h_o (minus must be punched into card)

DEPTH - the depth between the top surface and the bottom of problem. The units of DEPTH must correspond to the units in which the bubbling pressure head PB is given.

DELT - size of the dimensionless time step increments $\Delta\tau$ which are to be used in obtaining solution.

Q - dimensionless application rate $|v|/K_o$ X area of application.

PB - magnitude of the bubbling pressure head used to nondimensionalize all length parameters of the problem.

SSUR - If the upper surface boundary condition is to be used which specifies the application rate, SSUR must be given a value of zero. If the condition specifying the surface saturation is to be used SSUR equals the decimal surface saturation.

Card No. 2 (7F10.5)

SR - residual saturation S_r.

AMBDA - pore size distribution exponent λ.

POR - porosity η.

DTINC - not used in present program, but has been used in other versions to change time step $\Delta\tau$.

EXPAND - a parameter to expand the number of grid points at which values are computed at new time steps. When ξ changes from the initial conditions by an amount greater than EXPAND the number of grid points in either the radial or axial directions is expanded.

ERR - a parameter used to terminate the Newton-Relaxation method iteration. The individual line interations are terminated when the absolute sum of change between consecutive iteration is less than one-tenth of ERR.

VK - if application rate Q is to be specified, VK is given a value of zero. If the application flux $|v|/K_o$ is to be specified VK equals this value and the value assigned to Q is ignored.

Card No. 3 (4I5, F10.5)

NRIT1 - number of initial time steps (see INCB below) between which solution results are printed.

NRIT2 - number of regular time steps between which solution results are printed.

INCB - the first regular time step will be subdivided into INCB equal but smaller time steps. If the first time step is to be the regular size INCB = 1.

NHSTAR - if NHSTAR is less than zero only the values of the dependent variable ξ will be printed at the specified time steps. If NHSTAR = 0 values of ξ, the saturation and hydraulic head will be printed at the specified time steps. If NHSTAR is greater than zero, values of ξ will not be printed, but values of saturation and hydraulic head will be printed.

STREAD - if STREAD is less than 1.0 the initial condition will be generated within the program to satisfy the specified static equilibrium given by HI. Otherwise the initial condition is to be read in from input unit NWR2.

Card No. 4 (5I5)

NWRIT - Number of time steps for which solution results are to be stored on tape or other logical units denoted by NWR2. With present dimensions this must be 32 or less. If less than 1, results are output only on the system printer.

NWR2 - logical unit on which output is to be stored in addition to being output to the system printer. If initialization as input is called for, this is the unit for input also.

NSAT - not used in version listed, but in another version to eliminate printing of saturation or hydraulic head values.

NTABLE - not used in version listed, but in another version to generate table containing summary data.

MAX - maximum number of Newton-line iterations that will be allowed. The number of iterations on any line which will be allowed will be one-half this many.

No more cards are required for a solution which does not store solution results on another logical unit in addition to the system printer. If the solution results are to be stored on such a unit, one or more cards with the FORMAT 16I5 should contain the time step numbers for which the values of ξ are to be stored. To solve more than one problem at a time cards 1 through 4 (and possible card/or cards with time step numbers of stored solutions) are repeated.

The program listing contains some special binary tape or drum data transfer and manipulation routines available on the University of Utah UNIVAC 1108, but which are not standard FORTRAN. The names of these routines are: INOUT, SKPFLS, ENFILE and UNLOAD. If using a different system these names will likely need to be changed, these features of the program deleted or replaced by FORTRAN statements, or FORTRAN subroutines with these names added which perform essentially the same functions.